A MICROCOMPUTER SOLUTION TO MANEUVERING BOARD PROBLEMS

Kenneth Harper Kerns

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A MICROCOMPUTER SOLUTION
TO
MANEUVERING BOARD PROBLEMS

by

Kenneth Harper Kerns

and

Roger Stuart Cooper

Thesis Advisor: G. A. Kildall

June 1973

A Microcomputer Solution to Maneuvering Board Problems

by

Kenneth Harper Kerns
Lieutenant Commander, United States Navy
B.A., Vanderbilt University, 1961

and

Roger Stuart Cooper
Lieutenant, United States Naval Reserve
A.B., Princeton University, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1973

ABSTRACT

A special-purpose digital computer has been developed using the new MCS LSI microprocessor technology. The primary goal of this work was to solve a fairly complicated computing task using a minimal amount of random logic and limited development time. This computing system solves ships' maneuvering board problems including the determination of course, speed, and closest point of approach of other ships. Ten contacts can be tracked simultaneously. The triangulation involved in these calculations is performed using a decimal version of the CORDIC algorithm. A complete maneuvering situation can be computed in 4.5 seconds. Except for hardware required to drive the displays, all logic is contained in the software-encoded read-only-memory chips, which drive a single CPU chip. Approximately eight hundred man-hours were required to develop the programs and prototype the hardware. The system is modular and easy to maintain, low in bulk, power consumption, and cost.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF DRAWINGS

ACKNOWLEDGMENT

The assistance of three Naval Postgraduate School faculty members is gratefully acknowledged. Assistant Professor Gary A. Kildall served as both thesis advisor and chief scrounger, mysteriously acquiring all sorts of surplus and supposedly defective hardware. Instructor Ray H. Brubaker, second reader and remedial soldering instructor, demonstrated his infinite capacity for capacitors and small light bulbs in building the LED display and writing its associated software. And finally, Assistant Professor Richard H. Franke expended much of his time and more of his patience in premasticating several of the CORDIC papers.

Dei sub numine viget.

I. INTRODUCTION

A rapidly developing and expanding area of computer technology and application is that of microcomputers. Composed of miniaturized software-encoded chips and minimal hardware circuitry, the microcomputer is ideally adapted to specialized tasks and applications. It is the purpose of this thesis to demonstrate the applicability of the comparatively inexpensive microcomputer technology to Navy functions and problems.

An application which has been investigated by the Naval Electronics Laboratory Center, San Diego (NELC), is that of computerizing certain maneuvering board problems, in particular, determining the closest point of approach (CPA) of multiple radar-detected contacts. NELC has constructed a "breadboard version" of its CPA calculator, described in Refs. 1 and 2, making maximum use of off-the-shelf components. The heart of the unit is six MOS LSI chips manufactured by North American Rockwell Microelectronics Company and described in Refs. 3 and 4. They contain a standard array of arithmetic (add, subtract, multiply, divide) and scientific functions and constants (sine, cosine, exponent, natural logarithm, PI). The control program is stored in three Intel programmable read-only-memory (PROM) chips, with four Signetics random-access-memories (RAM) used for data storage. The NELC unit has a twenty key control keyboard and an eleven key (with decimal point) numeric keyboard, while the output is in the form of a light display of six digits plus sign. Information on five contacts can be stored simultaneously. Data input for each contact requires six individual entries unique to that contact, plus two which are the same for all contacts. Five pieces of output information for each

contact are computed and displayed, one at a time as directed by the user, with a total machine computation time for solutions less than sixteen seconds. NELC's method of solution involves manipulation of the input data using primarily the Law of Sines and the Law of Cosines. Inverse trigonometric functions are not available, and they are approximated by multiplicative constants, as indicated by Ref. 5. The NELC CPA calculator has been tested at sea [Ref. 5], and this test showed that a device which provides a quick, accurate, and reliable solution to maneuvering board problems will be eagerly received by potential Navy users.

The system described below has been developed as an alternate microcomputer solution to the CPA problem. The hardware is the Intel MCS-4 microcomputer set. The software functions have been limited to those which are required for the CPA solution. An extensive control program not only orders the various algorithms, but also makes sufficient checks on actions by the user to insure that input is complete and output will be meaningful. The system as implemented uses keyboard input and light display output, although alternate, more automated input/output schemes are suggested.

II. THE MANEUVERING BOARD

When a proceeding ship is near other ships, knowledge of the relationship of its motion to theirs becomes important in preventing collisions at sea. The Hydrographic Office has prepared a plotting sheet (H.O. 2665-10) called the Maneuvering Board which facilitates the solution of a ship's relative movement problem. This solution involves finding the closest point of approach of another ship along with its course and speed. Reference 6 was used as a guide in defining the terms and various plots on the Manuvering Board.

A. THE BASIC RELATIVE MOVEMENT PROBLEM

Figure 1 illustrates the format of a Maneuvering Board. Ship positions are plotted using the polar co-ordinate system. The center of the plot represents "reference" or "own" ship and any other point represents the position of a "maneuvering" ship, plotted in true bearing and range from own ship at various times. Solutions of relative motion problems with the Maneuvering Board consist of two distinct but related parts: the relative plot and the vector diagram, the latter often called the speed triangle.

1. The Relative Plot

In the relative plot the "maneuvering" ship is plotted at various times: its position at the beginning of the plot is labeled M1, and subsequent positions are labeled M2, M3, etc., as indicated in Figure 1. The line through M1, M2, M3, etc., is called the relative motion line, defining the direction of relative motion and the distance of relative motion. From this plot are obtained the bearing

and range of the maneuvering ship from own ship at any time and the bearing and range of the closest point of approach of the maneuvering ship from own ship.

The closest point of approach is indicated as the point labeled CPA in Figure 1. The range of CPA is simply the distance from the center of the plot to this point. The bearing of CPA is equal to the direction of relative motion plus or minus ninety degrees, dependent upon the quadrant containing the CPA point. The time of CPA is obtained by adding to the time of M1 the quotient of the measured distance from M1 to the CPA point and the speed of relative motion. This last computation can be accomplished through use of the logarithmic time, speed, and distance scale nomogram found at the bottom of the Maneuvering Board (H.O. 2665-10).

2. The Vector Diagram

The vector diagram is used to solve for the course and speed of the maneuvering ship. The vector diagram is a triangle composed of the three vectors shown in Figure 1. Vector er indicates the course (vector direction) and speed (vector magnitude) of own ship, while Vector em represents the course and speed of the maneuvering ship. Vector rm represents the direction and speed of relative motion. Thus, given own ship's course and speed and the direction and speed of relative motion determined from the relative plot, the course and speed of the maneuvering ship is found by drawing vector em to complete the triangle erm.

MANEUVERING BOARD SCHEMATIC

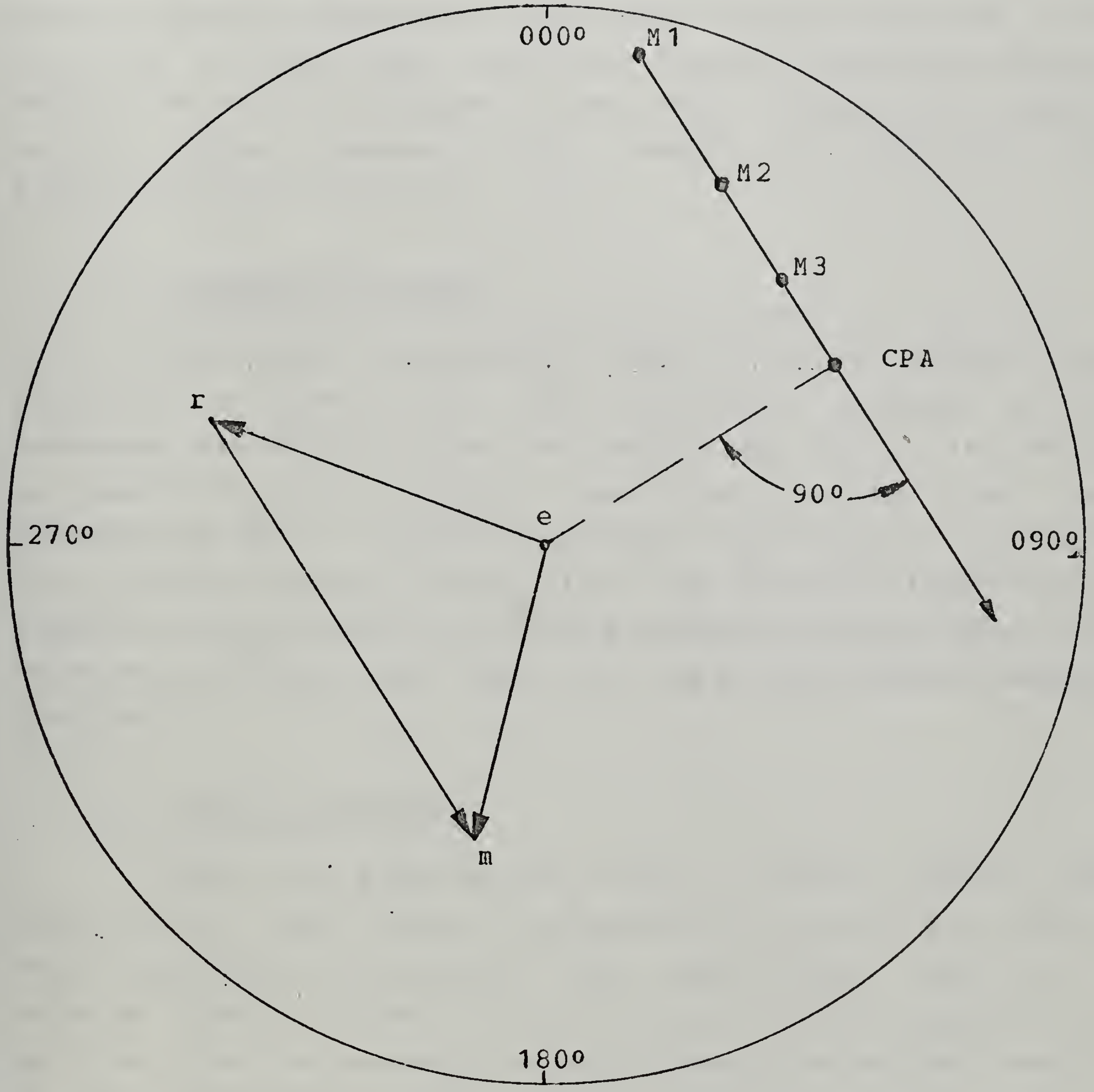


FIGURE 1

B. OTHER USES OF THE MANEUVERING BOARD

All maneuvering board problems utilize the basic relative motion problem discussed above. In addition to determining CPA information, the Maneuvering Board can also be used to find the required course and speed to take station on or to intercept another ship, to find true wind, or to find courses and speeds for scouting and torpedo-firing situations.

1. Stationing Problem

Given the maneuvering ship's course, speed, and position at some time, the stationing problem is to determine own ship's course, or course and speed, in order to maneuver to a certain range and bearing from the maneuvering ship. The basic solution technique is to draw the relative motion line from the starting maneuvering position to the point of required range and bearing from the maneuvering ship, and then to solve the relative motion problem.

2. Intercept Problem

Given the maneuvering ship's course, speed, and position at some time, the problem is to find own ship's course and speed to intercept the maneuvering ship by a certain time. To find a solution, the relative motion line is drawn from the maneuvering ship's position to the center of the maneuvering board, after which the relative motion problem is solved.

3. True Wind Problem

Since the known apparent wind is the resultant of ship's motion and true wind, the maneuvering board can be

used to solve for true wind. The solution for true wind is simply the em vector of the vector diagram, where ship's motion is the er vector and the measured apparent wind is the rm vector.

4. Military Applications

The Maneuvering Board is used for other problems such as scouting and torpedo problems. These problems involve determination of courses and speeds to undertake a desired scouting or search path, or a course and speed required to shoot a spread of torpedoes. Dutton [Ref. 6] provides descriptions and solution techniques for these problems as well as the problems outlined above.

III. CORDIC COMPUTING TECHNIQUE

The maneuvering board algorithm described below is based upon a vector analysis of the data, which in turn requires certain transcendental functions: sine, cosine, inverse tangent, and modulus. Sine and cosine generating techniques had already been implemented on the MCS-4, as described in Ref. 7, with average run times of 650 and 750 milliseconds, respectively. It was recognized that these routines would yield an unexceptable total computation time if used in the maneuvering board algorithm. Consequently, the CORDIC algorithm described below was investigated and eventually implemented, significantly reducing the time required for each function.

The CORDIC (CO-ordinate Rotation DIgital Computer) technique was developed by J. E. Volder in 1959 [Ref. 8]. It is a simple, accurate, and relatively fast method for generating transcendental functions involving only the operations of addition, subtraction, and shifting.

The principle involved in CORDIC is to rotate an imaginary vector represented by its x,y components through a sequence of angular steps of predetermined magnitude. At each step the values of the angle and the x,y components of the vector are modified according to a set of equations that can be implemented with add, subtract, and shift operators.

A. CORDIC PRINCIPLES

J. S. Walther has described [Ref. 9] a single unified algorithm for the calculation of elementary functions using co-ordinate rotation in a linear, circular, or hyperbolic co-ordinate system. For the sake of simplicity, the description of the CORDIC technique is based upon the

circular co-ordinate system.

1. Iteration Equations

Consider a vector $P = (x,y)$ with radius $R = (x^2 + y^2)^{1/2}$ and angle $A = \tan^{-1} y/x$. Let a new vector $P_{i+1} = (x_{i+1}, y_{i+1})$ be obtained from $P_i = (x_i, y_i)$ by rotating the co-ordinates by $\tan^{-1} D_i$ according to

$$x_{i+1} = x_i \pm y_i D_i \tag{1}$$

$$y_{i+1} = y_i \mp x_i D_i . \tag{2}$$

The $(\pm)$ and $(\mp)$ signs indicate that $y_i D_i$ and $x_i D_i$ are either added to or subtracted from the first term depending on the polarity of $\tan^{-1} D_i$, in which D_i is an arbitrary value that will be described below.

The angle and radius of the new vector P_{i+1} in terms of the old vector P_i are

$$A_{i+1} = A_i - \tan^{-1} D_i$$

$$R_{i+1} = R_i (1 + D_i^2)^{1/2} ,$$

where $(1 + D_i^2)^{1/2}$ represents the elongation factor of the magnitude of the vector after each co-ordinate rotation.

For n iterations:

$$A_n = A_0 - a$$

$$R_n = R_0 K,$$

where

$$a = \sum_{i=0}^{n-1} \tan^{-1} D_i$$

and

$$K = \prod_{i=0}^{n-1} (1 + D_i)^{1/2}.$$

Let a third variable Z be provided for the accumulation of the angular variations $\tan^{-1} D_i$ so that

$$Z_{i+1} = Z_i + \tan^{-1} D_i. \qquad (3)$$

Solving the set of equations (1), (2), and (3) for n iterations, the final values are

$$X_n = K(X_0 \cos a + Y_0 \sin a)$$

$$Y_n = K(Y_0 \cos a - X_0 \sin a)$$

$$Z_n = Z_0 + a.$$

2. Convergence Scheme

The angle A of vector P can be forced to zero by a converging sequence of rotations, $\tan^{-1} D_i$, which at each step brings the vector closer to the positive x-axis. The direction of rotation must be determined at each step such

that

$$|A_{i+1}| = | |A_i| - \tan^{-1} D_i |. \qquad (4)$$

The same idea applies to forcing the angle in Z to zero. It can be shown that A or Z converges to within $\tan^{-1} D_{n-1}$ of zero in n iterations [ref. 9].

3. Shifting

The CORDIC technique requires the use of shifting to effect the multiplication in equations (1) and (2). Let $D_i = r^{-i}$ where r is the radix of a number system and i is any positive integer; then a multiplication of x by D_i is a shift of x by i places to the left. The integer i must be chosen such that the angle $\tan^{-1} r^{-i}$ satisfies the convergence criterion discussed above. Note that the accuracy at the n-th iteration is approximately equal to n-1 places [ref. 9].

4. CORDIC Computing Process

In the computing process iteration equations (1), (2), and (3) are used, while the direction of rotation is determined by equation (4). The $\tan^{-1} D_i$ constants are preloaded in either degrees or radians for i=0,1,2,3,....,n.

There are two computing modes, rotation and vectoring. In the rotation mode Z is driven to zero, and in the vectoring mode A is driven to zero; that is, $Y_n = 0$.

With the circular co-ordinate system as a basis, entering X = 1/K, , Z = some angle θ, and using the rotational mode, the results will be

$$X = \cos \theta$$
$$Y = \sin \theta$$
$$Z = 0,$$

which are accurate to approximately n-1 places. In the vectoring mode entering $X = x_0$, $Y = y_0$, Z = 0 leads to

$$X = K(x_0^2 + y_0^2)^{1/2}$$
$$Y = 0$$
$$Z = \tan^{-1} y_0/x_0,$$

accurate to approximately n-1 places.

Using the linear or hyperbolic systems as a basis to the CORDIC technique, multiplication, division, square root, hyperbolic sine and cosine, or inverse hyperbolic tangent functions are obtained. The remaining transcendental functions can be computed from mathematical identities involving the CORDIC-derived functions. Reference 9 supplies more information in this area.

B. APPLYING CORDIC TECHNIQUE TO THE MCS-4

In order to solve the basic relative movement problem of the maneuvering board, conversions between the polar and rectangular co-ordinate systems are required. The CORDIC technique is ideally suited for this task since it provides the sine and cosine values of an angle in the rotation mode and $(x^2+y^2)^{1/2}$ and $\tan^{-1} y/x$ in the vector mode.

1. Decimal Implementation

The MCS-4 prototype uses the fixed point decimal number system. The CORDIC technique was developed in fixed point decimal form to facilitate the required shifting operation, since in this case shifting is, in effect, a relocation of the decimal point.

The radix of the decimal system is ten, thus D_i was chosen as 10^{-i} and angular steps of $\tan^{-1} 10^{-i}$ are used. In order to obtain a desired four digit accuracy, five angular steps were required. The values of $\tan^{-1} 10^{-i}$ decrease by approximately 1/10 whenever i is increased by one.

To cover all angles between zero and ninety degrees, several subiterations were required for each angular step, as outlined in Table I.

TABLE I

CORDIC ARCTANGENT CONSTANTS AND SUBITERATIONS

i	$\tan^{-1} 10^{-i}$	Number of Subiterations
0	45°	1
1	5.7105931°	7
2	0.5729386°	9
3	0.0572957°	9
4	0.0057295°	10

The number of subiterations j for each angular iteration i was determined from the inequality

$$\sum_{j} \tan^{-1} 10^{-(i+1)} \leq \tan^{-1} 10^{-i}.$$

This was done to minimize the total number of iterations, thus reducing the execution time of the MCS-4 prototype.

For any angular step numbered four or greater, the number of subiterations will be ten. Values of $\tan^{-1} 10^{-i}$ were preloaded into ROM in degrees. It should be pointed out that angular values could have been represented in radian units, but degrees were chosen to allow the user to enter information without the need for conversion.

2. Rotation Mode

In the rotation mode the sine and cosine of an input angle ranging from zero to 360 degrees is computed. The flow chart for this process is shown in Appendix D.

The initial values are:

$$X = 1/K = 0.682594516$$

$$Y = 0$$

$$Z = \theta.$$

K represents the elongation factor which is precomputed from the formula

$$K = \prod_{i=0}^{4} (1 + 10^{-i})^{j/2},$$

where j represents the number of subiterations for each i as in Table I above.

Prior to the first angular iteration of forty-five degrees, the input angle is tested for polarity and the x,y components rotated plus or minus ninety degrees to place the angle in the first or fourth quadrant. This is done so that

at the next rotation of plus or minus forty-five degrees, the angle will be within forty-five degrees of the positive x-axis.

At each angular iteration, Z is tested and driven to zero; that is, if Z is positive, the rotation is done in the negative direction, or if negative, rotated in the positive direction. The iteration equations are then carried out by either addition or subtraction, and by a shift of i decimal places to the left. After the final iteration

$$X = \cos Z$$
$$Y = \sin Z$$
$$Z = 0.$$

3. Vectoring Mode

In the vectoring mode the y component of the pseudo vector is tested and driven to zero. This test is equivalent to driving the vector angle A to zero. The flow chart for this process is shown in Appendix E. The appropriate values are entered for X and Y with Z equal to zero. The output is

$$Z = \tan^{-1} Y/X$$

$$X = K(X^2 + Y^2)^{1/2}$$

$$Y = 0.$$

There are two tests made prior to the first angular iteration. The first test is to compare the input values of Y and X. If the absolute value of Y is greater than the absolute value of X, then the x,y coordinates are rotated ninety degrees in a neagative direction. This procedure will keep the truncation error of Y/X to a minimum. Following the test X is examined to insure that it is positive. If it is not positive, the x,y coordinates are

rotated 180 degrees to make X positive.

After the final iterations are done, the value of X is multiplied by 1/K in order that X will finally contain $(X^2+Y^2)^{1/2}$, where K is the elongation factor previously described.

C. OTHER IMPLEMENTATIONS OF CORDIC

Two other approaches to the CORDIC technique are briefly discussed. They are a decimal approach proposed in Ref. 10 and an approach described in Ref. 11 commonly used for hand electronic calculators including the Hewlett-Packard HP-35.

1. Decimal Approach

The CORDIC technique has primarily been implemented in binary form. Schmid and Bogacki [ref. 10] demonstrate a method of using the CORDIC technique in decimal form. For all angular steps after the first, they suggest using the constant number of substeps, nine. This simplifies the programming of the algorithm, but as a result it uses a total of thirty-seven steps to compute trigonometric functions to five-digit accuracy. Using a constant number of substeps is in contrast to the decimal implementation described above. there the number of substeps was minimized, constrained by the convergence criterion and the desired n-1 place accuracy.

2. HP-35 Approach

In many small calculators compactness of code is more important than speed. In the HP-35 Calculator the CORDIC algorithm is used to simultaneously generate K sin Z and K cos Z where K is the elongation factor. Tan Z is derived from these by division and then sin Z and cos Z are derived using identities involving square root and tan Z.

The square root function uses the conventional sum of odd digits algorithm. The advantage of this approach is the lack of any need to use predetermined elongation constants, which is important in a decimal machine where the number of subiterations and the sequence of rotation constants is a function of the data [Ref. 11].

IV. IMPLEMENTATION

Software implementation is divided into two major areas, the maneuvering board solution algorithm and the control program. The maneuvering board solution algorithm is the heart of the software system. To be executed sucessfully, the maneuvering board algorithm must have inputs which are complete and within certain bounds. An extensive control program is required to protect the user from misuse and to insure all necessary input values are supplied before the maneuvering board solution algorithm can be called. Additionally, the control program facilitates storage of observed data for multiple contacts.

A. HARDWARE CONSIDERATIONS

The concept of solving Navy problems with microcomputers is in no way dependent upon the capabilities and characteristics of a particular manufacturer's product. Several factors, however, led to the selection of the Intel MCS-4 Micro Computer Set as the prototype hardware for the maneuvering board computer. The computer set is relatively inexpensive and was available at the Naval Postgraduate School, along with a good library of associated utility software and an interactive assembler/simulator running under CP/CMS on the School's IBM 360/67, as described in Ref. 12. The Intel product operates using programmable read-only-memory (PROM), allowing construction of a functioning prototype without the added expense of ROM etching. After selection of the MCS-4, several implementation decisions made during development of the system were directly dependent upon hardware features of the set.

The CORDIC algorithm is essential to the implementation of the maneuvering board solution described below. CORDIC, in turn, requires an efficient shifting operation. The MCS-4 is a hexadecimal machine: binary shifting is cumbersome and time consuming. For this reason, and because input by the user is in the form of decimal digits, numbers are stored and manipulated in decimal character form through use of the machine's decimal adjust instruction. The random-access-memory (RAM) of the MCS-4 is organized and addressable in sixteen-byte blocks or registers. For this reason, and because all values involved in computation fall within a narrow enough range, fixed point representation of numbers was chosen as the simplest and least time consuming for manipulation. The largest input value in any maneuvering board problem is a contact's range, practically limited by search radar to approximately one hundred thousand yards. All numbers, therefore, are represented as sixteen-digit decimals in tens-complement form, seven digits before the decimal point and nine after.

Reference 13 is a complete description of the MCS-4 hardware and a full explanation of its instruction set.

B. MANEUVERING BOARD SOLUTION ALGORITHM

The maneuvering board algorithm was designed to solve the basic relative movement problem described above. With this algorithm closest point of approach problems, stationing problems given time to station, and intecept problems can be solved. The algorithm utilizes conversion between polar and rectangular co-ordinates, together with vector addition and subtraction to solve the basic maneuvering board problem. The decision to use this method rather than using the Law of Cosines and the Law of Sines was based upon the realization that the CORDIC algorithm neatly provides the cosine and sine of an input angle in the

rotational mode and $(x^2+y^2)^{1/2}$ and $\tan^{-1} y/x$ in the vector mode. These functions are necessary in conversion from polar to rectangular co-ordinates and back again. The flow chart for the maneuvering board solution algorithm is found in Appendix F.

1. Solution of the Relative Plot

Let the following be inputs: own ship's course, CO, speed, SO, and two positions in time; that is, bearing, B1, B2, range, R1, R2, and time, T1, T2. The first step is to convert the positions to rectangular co-ordinates:

$$X1 = R1 \cos B1$$
$$Y1 = R1 \sin B1$$
$$X2 = R2 \cos B2$$
$$Y2 = R2 \sin B2.$$

Considering X1, Y1 and X2, Y2 as the x,y components of two vectors, V and W respectively, the relative motion line can be considered as the vector derived from the subtraction of vector V from vector W. Thus the x,y components of the relative motion line are

$$X0 = X2 - X1$$
$$Y0 = Y2 - Y1.$$

Since the quotient Y0/X0 represents the slope of the relative motion line, Y0 and X0 are compared to avoid a possibly infinite slope. If |Y0| is greater than |X0|, then the x,y components are rotated ninety degrees and the slope M = Y0/X0 is computed. The direction of relative motion (DRM) and the speed of relative motion (SRM) are then found by converting the x,y components of the relative motion line back to polar co-ordinates.

$$DRM = \tan^{-1} (Y0/X0)$$

$$SRM = \frac{(XC^2 + YC^2)^{1/2}}{T2 - T1}$$

The closest point of approach can be solved for in rectangular co-ordinates as the intersection of the relative motion line having slope M and passing through points (X1,Y1) and (X2,Y2) and a line perpendicular to the relative motion line which passes through the origin and has slope -1/M. Specifically,

$$XCPA = \frac{X1\ M^2 - Y1\ M}{M^2 + 1}$$

$$YCPA = \frac{Y1 - X1\ M}{M^2 + 1}.$$

The rectangular co-ordinates of the closest point of approach are then converted to polar co-ordinates, yielding the range and bearing of CPA

$$\text{CPA Range} = (XCPA^2 + YCPA^2)^{1/2}$$

$$\text{CPA Bearing} = \tan^{-1} (YCPA/XCPA),$$

where the time of CPA is

$$\frac{[(XCPA-X1)^2 + (YCPA-Y1)^2]^{1/2}}{SRM} + T1.$$

2. Solution of the Speed Triangle

Considering that the speed triangle is a triangle of three vectors and that the maneuvering ship's course and speed represent one of these vectors, then the maneuvering ship's course and speed can be found as the resultant of the other two.

The basic ideas described above are used again except that speed replaces distance as the magnitude of the vectors. The polar forms of the two known vectors are transformed to rectangular co-ordinates.

Given own ship's course (CO), own ship's speed (SO), speed of relative motion (SRM), and direction of relative motion (DRM), then

$$X1 = SO \cos CO$$
$$Y1 = SO \sin CO$$
$$X2 = SRM \cos DRM$$
$$Y2 = SRM \sin DRM.$$

Thus the x,y components of the maneuvering ship's vector are X1 + X2 and Y1 + Y2 where the maneuvering ship's speed is

$$[\ (X1 + X2)^2 + (Y1 + Y2)^2\ ]^{1/2}$$

and the course is

$$\tan^{-1} (Y1 + Y2 \ / \ X1 + X2).$$

C. THE CONTROL PROGRAM

The CPA algorithm and other routines invoked as required by the actions of the user are controlled by and imbedded

within a constantly executing, continuously looping control program. The flow of control is determined by the user's depressing one of twenty-five keys. The control loop constantly queries the keyboard input until a key depression is detected, and then branches program control to a corresponding section of code. Upon completion of the selected function, control returns to the keyboard query loop.

Although only five bits are required to specify which key has been depressed, the control program is written for a keyboard with a six bit output, the sixth bit being on when and only when a key is actually depressed. The control program therefore reads two four-bit ROM input ports in sequence, the first containing the indicator bit and one identification bit, the second containing the four remaining identification bits. The five identification bits are used as an index into a table of branch instructions to routines corresponding to the keys.

The control program performs two additional functions. The query loop, while waiting for a key to be depressed, repeatedly invokes a subroutine to display the current digital output. After a key has been depressed, the control loop zeroes the information on a particular ROM output port. One bit on this port is used to indicate to the user an incorrectly selected key. The key routines are heavily interactive, and any key depressed out of proper sequence will cause this bit to be turned on, activating an error light.

Details of the control program's logic are provided in Appendix B.

D. THE DISPLAY INTERFACE

Output to the user is provided in two forms: the error light previously described and a sixteen digit light emitting diode (LED) display. The routine controlling this

display is dependent upon the hardware used. The interface to this routine, however, is determined independently. Prior to invoking this routine, the information to be displayed must be placed in a RAM register in the following manner: the rightmost digit of the display must be the low order digit (digit zero) in the RAM register, and correspondingly the leftmost digit must be in the high order location (digit fifteen). The decimal points of the LED's are lighted by the display routine according to the contents of the sixteen bits of the RAM register's four status characters. The leftmost bit of status character zero being on indicates that the decimal point in front of the low order digit is to be lighted. Correspondingly, the rightmost bit of status character three controls the decimal point in front of the high order character. Finally, the address of the RAM register to be displayed must be placed in the particular CPU index register pair required by the display routine. Since the display hardware is assumed to have no memory, the display routine must be invoked over and over at a rapid rate to achieve sufficient brightness and to avoid flicker.

The particular display routine for the prototype machine is available in the program listing, annotated with an explanation of the basic hardware and how it is used. Another display scheme is described in Ref. 14, included in which is a brief explanation of the problems involved in MCS-4 input/output.

V. CONCLUSION

In designing the software for the MCS-4 prototype, the three most important considerations were execution time, memory size, and user interaction. As a result, the maneuvering board algorithm was written to solve the basic relative motion problem only. Since all of the maneuvering board problems use this basic solution, or at least part of it, the algorithm provides the solution foundation for all related problems.

A. RESULTS

Presently the MCS-4 prototype will handle all input information for ten simultaneous contacts, and it can be used to solve the CPA problem, the stationing problem given time to station, and the intercept problem. These problems are solved in 4.5 seconds of execution for all five items of output information. This relatively fast solution time is largely due to the speed of the transcendental CORDIC functions developed for this computer as described above: the rotation mode yields sine and cosine in approximately 360 milliseconds, while the arctan/modulus function takes about 410 milliseconds. Additionally, output information is displayed, not one item at a time, but in sets of associated information: the course and speed comprise one set, and the CPA range, bearing, and time, the other.

As a demonstration of the capability of this prototype, several example problems were worked on the Maneuvering Board (H.C. 2665-10) and compared with the prototype's output. Differences between the two methods of solution are less than one degree in course or bearing, one minute in time, one hundred yards in range, and one-half knot in

speed. The accuracy of the microcomputer's outputs has also been verified through comparison with the answers provided by a FORTRAN simulation of the maneuvering board algorithm written for debugging purposes during implementation of the prototype. Table II illustrates comparisons for three example CPA problems.

Three additional points require explanation. For CPA solutions where both maneuvering ship positions are exactly the same but at different times, the MCS-4 prototype solution appears to be in error as to the closest point of approach. In this special situation the solution is simply that the maneuvering and reference ship are on the same course and speed and that the maneuvering ship is constantly at CPA. The prototype's software was designed to compute the CPA point as the intersection of two straight line, and therefore will not handle a case where one of the lines is actually a point. The maneuvering board algorithm could be modified to handle this situation, but since this rarely occurs, and since the solution is immediately apparent to the trained operator, it was not incorporated.

The second point pertains to the inability to solve a stationing problem given speed to station. This problem requires solution for the x,y components of the maneuvering ship's vector where the magnitude (speed) of the maneuvering ship's vector is known, but the magnitude of the relative motion vector cannot be obtained using the methods described above. The maneuvering board algorithm finds the speed of relative motion by using the time between observations as a divisor. Since the time on station is not an input value, this time difference is not available. The comparable manual solution technique requires swinging an arc of radius equal to the stationing speed to find its intersection with the relative motion vector, thereby defining the rm vector.

Finally, the ability to solve true wind problems could have been incorporated in the prototype, but the need for additional control logic precluded it in the present

TABLE II

CPA PROBLEMS

	PROBLEM ONE	PROBLEM TWO	PROBLEM THREE
REFERENCE SHIP			
Course	350°	000°	025°
Speed	15 kts	14.5 kts	20 kts
MANEUVERING SHIP			
Position 1			
Time	0910	1400	1400
Bearing	012°	103°	340°
Range	27,600 yds	14,000 yds	27,000 yds
Position 2			
Time	0920	1429	1406
Bearing	000°	148°	330°
Range	22,000 yds	14,000 yds	18,800 yds
MANEUVERING BOARD SOLUTION			
CPA Bearing	319°	(past CPA)	271°
CPA Range	16,500 yds		9,800 yds
CPA Time	0939		1416.8
Course	269°	311°	165°
Speed	20 kts	8.5 kts	28.5 kts
MCS-4 PROTOTYPE SOLUTION			
CPA Bearing	318.9°	(past CPA)	271°
CPA Range	16,594 yds		9,695 yds
CPA Time	0938.9		1416.8
Course	269.3°	310.3°	164.4°
Speed	19.8 kts	8.4 kts	28.3 kts

implementation. True wind problems require no relative plot, and hence would utilize only that portion of the maneuvering board algorithm dealing with the speed triangle.

Examples of stationing given time to station and intercept problems are provided in Appendix A, the users' guide to the prototype machine. Cost estimates are presented in Appendix C.

B. FUTURE WORK

The maneuvering board computer can be a valuable tool to Navy users as implemented. Several extensions are possible, however, in both software and hardware.

Certain software features have not been implemented for lack of time. The ability to display data previously entered could be added most quickly. The ability to compute the reciprocal of a bearing may be of use while entering data in a stationing problem, where own ship's bearing from another ship is required. A more significant improvement would be the ability to compute stationing course given stationing speed instead of time to station, but this requires a second solution algorithm with different inputs as described previously.

The system can be modified to handle more than ten contacts. This would require reallocating one of the sixteen display digits to the contact identification number. Each additional contact requires two additional RAM registers, or half of a RAM chip.

Alternate hardware methods of input/output could significantly improve the maneuvering board microcomputer as a total system. The present keyboard input could be largely supplanted by microcomputer-controlled analog-to-digital converted outputs taken directly from the radar's cursor circuits and a real-time clock. Similarly, output could take the form of a video display whose input is prepared by a microcomputer-controlled digital-to-analog conversion of

the presently available digital output. These features could be implemented inexpensively using independent microprocessors; the maneuvering board computer, handling a more complex problem, requires only thirteen ROM chips, nine RAM's, and a single-chip CPU.

The results of this thesis indicate that microcomputer technology can feasibly be applied to the ideas above and to similar Navy problems.

APPENDIX A

USERS' GUIDE TO MANEUVERING BOARD COMPUTER

This guide to the MCS-4 prototype maneuvering board computer is divided into two sections. The first part consists of instructions to the user and a description of the interactive responses the user may expect in solving CPA problems. The second section contains examples of other types of maneuvering board problems and directions on how the prototype computer can be used for their solution.

A. THE BASIC CPA PROBLEM

In all maneuvering board problems the identification of both a reference ship and a maneuvering ship is required. In a CPA problem the reference ship is the user's own ship, and the maneuvering ship is a contact whose range and bearing are known.

When the machine is first energized, the reinitialization button located on the body of the MCS-4 should be depressed. Then, or at any other time desired by the user, this button erases all stored information, blanks the display, and restarts the internal program.

In the instructions below, keys are identified by the symbols used in Figure 2, the keyboard/display organization schematic. The locations of the various information items in the LED display are indicated in Figure 2 by these same key indentification symbols.

There are two types of data which must be entered and stored prior to any solution. The reference ship's data consists of a course and speed. The maneuvering ship's data consists of an identification number, time, bearing, and range for each of two observations.

Certain instructions are common for all six items of input information. Depressing the item's input key terminates the entry of any previous item and enables the number keys; there is no visible interaction. For each item at least one number must be entered before a different input key can be depressed. An incorrectly entered item can be erased from the display by depressing that item's input key again. For each item there is a maximum number of digits which can be entered. All but range and contact identification number can have an additional digit after a decimal point. After at least one digit has been entered, the decimal point key may be depressed, followed by one more digit. Although the user has displayed a decimal point, a digit does not have to follow; a zero will be assumed.

To enter own ship's information, first depress either the speed (S) or course (C) key: these items may be entered in either order. Speed may have up to two digits to the left of a decimal point, course three. Speed is measured in

PROTOTYPE KEYBOARD/DISPLAY SCHEMATIC

LED DISPLAY

						C	C	C	.C				S	S	.S
N	T	T	T	T	.T	B	B	B	.B	R	R	R	R	R	R

KEYBOARD

	CLEAR KEY	
7	8	9
4	5	6
1	2	3
0		.

CPA	TS	
IST	SAVE	N
2ND		T
OS	C	B
COMPUTE	S	R

FIGURE 2

knots, course in degrees between zero and 360. Leading zeroes are assumed. When course and speed have not yet been entered after a reinitialization, both must be entered at the same time. When course and speed appear correctly on the display, depress the own ship (OS) key. The display will then be blanked, indicating proper storage. Note that once either the C or S key has been depressed, the N, T, B, and R keys cannot be depressed without error until the OS key has been depressed successfully or until the CLEARKEY key is depressed. CLEARKEY clears the display whenever it is depressed, thus erasing any entries which have not been stored. At any time a key is depressed out of acceptable sequence, the error indicator will light and remain lit until a vaild key selection is made.

Contact number (N), time (T), bearing (B), and range (R) keys function similarly to C and S. N will take only one digit, between zero and nine. T takes four digits to the left of a decimal point, measuring time on a twenty-four hour clock. The digit after the decimal point represents tenths of minutes and is optional. B takes three digits, again measured in degrees from zero to 360. R takes up to six digits, and is measured in yards. Leading zeroes are assumed. These four items may be entered in any order, and once all four are correctly displayed, they may be stored by depressing either the first (1ST) or second (2ND) observation key, as appropriate. Note that depressing 2ND will generate an error and will not blank the display if a first observation has not already been stored for the contact whose number appears in the display. Again, once one of these four keys is depressed, C and S cannot be depressed without error until 1ST, 2ND, or CLEARKEY has been successfully selected.

To compute CPA information for a contact, first enter only the contact's identification number in the display, then depress the COMPUTE key. If two observations for this

contact and the reference ship's data have been stored previously, the display will be blanked, and after 4.5 seconds the contact's number will reappear along with its time, bearing, and range of CPA. If the time of the contact's CPA is earlier than the time of its second observation, the contact is past CPA and the display will remain blank except for two decimal points. In either case depressing the "targetship" (TS) key will display the contact's course and speed. The CPA key will return the CPA output data to the display. Since CLEARKEY merely erases the display, TS and CPA can be successfully selected after CLEARKEY.

After computation on a contact, either the first or second observation data can be saved to be used as the first observation for a later CPA calculation. Depress either 1ST or 2ND, as appropriate, followed by SAVE. 1ST or 2ND after a COMPUTE, and only then, will blank all of the display except the contact number and display a one or a two as the rightmost digit. SAVE will blank the display and store the designated data as a new first observation. TS and CPA remain enabled. After a COMPUTE, depressing C, S, N, T, B, or R will clear the display and disable CPA, TS, 1ST, 2ND, and SAVE.

Any stored contact observation may be overwritten by the user in the manner of a normal entry described above. After own ship's course and speed have once been entered following a reinitialization, then only one or the other need be entered if either must be changed. In this circumstance, however, all contact observation data is erased. To obtain valid solutions, all observations and computations must be made for a reference ship on a constant course and speed. As a general rule, if observation data and new reference ship data are available for entry at the same time, enter reference ship's data first.

The computer will store data for up to ten contacts. Solutions for the ten contacts may be computed in any order.

E. EXAMPLES OF USE OF THE MCS-4 PROTOTYPE

As indicated above the MCS-4 prototype is primarily used for CPA solutions, but it can also be used for intercept, stationing given time to station, and collision avoidance problems. The sample problems below are used to illustrate how the MCS-4 prototype can be employed.

1. Intercept Problem

Consider a ship contact which is on a course of 200° at a speed of 10 knots. Its bearing and range at time 1010 are 340° and 15,000 yards from own ship. At time 1010 maneuver own ship to intercept the ship contact at time 1100.

The solution procedure is to consider the ship contact as the reference ship and own ship as the maneuvering ship. Enter course 200°, speed 10 knots using the C, S, and OS command keys. Since the contact is the reference ship, enter the reciprocal of the first bearing, 160°, the range of 15,000 yards, the time of 1010, and the contact number. Command keys B, R, T, N, and IST are used. Next enter as the second position a bearing of 160°, a range of one yard, the time of 1100, and the same contact number. Command keys B, R, T, N, and 2ND are used.

To compute the solution, enter the contact number and depress the COMPUTE key. A CPA of zero yards at time 1100 will be displayed. For the intercept course and speed depress the IS command key. The course and speed of 261.7° and 6.5 knots will be displayed.

A solution of 262° and 6.6 knots was obtained when working the above problem on the Maneuvering Board.

2. Stationing Given Time to Station Problem

Consider a ship contact which is on a course of 160°

at a speed of 18 knots. Its bearing and range from own ship are 020° and 16,000 yards at time 1320. At 1320 maneuver own ship to take station so that the ship contact bears 070° at a range of 2,000 yards at time 1350.

Again the solution procedure is to consider the ship contact as the reference ship and own ship as the maneuvering ship. Enter a course of 160° and a speed of 18 knots. Since the ship contact is now the reference ship enter the reciprical bearing 200°, range 16,000 yards, time 1320, and the contact number for the first position. Next enter the reciprical bearing 250°, range 2,000 yards, time 1350, and the same contact number for the second position.

For solution results, enter the contact number and depress the COMPUTE command key. Once the CPA results are displayed depress the TS command key to obtain the course and speed to station. The course and speed of 104.7° and 10.0 knots will be displayed.

A solution of 104° and 10.1 knots was obtained when working the above problem on the Maneuvering Board.

3. Collision Avoidance Problems

Collision avoidance problems are situations where a ship contact is determined to be holding a steady bearing or a very slight bearing drift with a decreasing range. In this situation the MCS-4 prototype can be used to determine a course and speed which will open the CPA range. If some specified time is given, for example the time the collision is expected to occur, the collision avoidance problem is exactly a stationing problem given time to station. The ship contact is treated as the reference ship and its reciprocal bearings are entered. The second position data is entered as the desired closest point of approach of the ship contact at the expected collision time.

Note that in solving any of these non-CPA problems, entering a contact's course and speed as the reference

ship's data will, for the reasons described above, erase all the previously entered observation data for other contacts.

APPENDIX B

CONTROL PROGRAM LOGIC DESCRIPTION

the control program is described below in terms of its logical flags and counters and the functions performed when each key is depressed by a user.

A. FLAGS AND COUNTERS

Before a CPA solution can be determined for a particular contact, eight separate inputs must be stored in the computer. The flags and counters described below are used to guide the input of information and the execution of the CPA algorithm.

1. Format Flags

Inputs to the machine are made in one of two formats. The first corresponds to the four pieces of information required to represent the observation of a contact: the contact's identification number, the time of observation, its bearing, and its range. The second format is used for entering the reference ship's course and speed. There is a flag corresponding to each format, set when information making up that format is being entered.

2. Selection Flags

When information is being input to the computer, one of six selection flags will be set, corresponding to the particular item being entered.

3. Input Flags

Six input flags correspond to the six selection flags. These flags are set when a sufficient number of digits for the current item has been entered.

4. Storage Flags

A flag is set when the contents of a complete format is entered into storage. One storage flag indicates that the reference ship's course and speed have been entered. The twenty other storage flags, two for each of ten possible contacts, are used to indicate the storage of first and second observations' data.

5. The Decimal Point Flag

A decimal point may be requested only once for a given input item, and a flag is set when this request has been made.

6. The Solution Flag

When the CPA algorithm is executed, a solution flag is set indicating that output information is available for display.

7. Counters

Two CPU registers are used as counters indicating the maximum number of digits allowed and the minimum number required for a particular input.

B. THE KEYBOARD

Twenty-five functional and numeric keys are required for control of and input to the computer.

1. Input Keys

There are six input keys corresponding to the six types of input information: contact identification number, time of observation, its bearing and range, and the reference ship's course and speed. Each key performs the following functions:

a. Each verifies that the format flag for the format not corresponding to the key is not set. Otherwise, it indicates an error and returns to the control loop.

b. Each then verifies that the minimum number of digits for the preceding input item has been supplied, or ascertains that the preceding item is also the item just selected. Otherwise, it indicates an error and returns to the control loop.

c. The key then resets all selection flags, the decimal flag, and the input flag corresponding to the item selected. It sets the format flag and the selection flag corresponding to the item selected. If the solution flag is set, it resets that flag and blanks the entire display; otherwise, it blanks that portion of the display corresponding to the item selected.

d. Finally, it initializes various CPU registers as required to prepare the display for the digits which will follow.

2. Storage Keys

The three storage keys correspond to the first and second observations of a contact, and to the reference ship's course and speed data. The following common functions are performed by each of the three keys:

a. Each verifies that all input flags required by the corresponding format are set. Otherwise, it indicates an error and returns to the control loop.

b. It stores the information, setting the appropriate storage flag.

c. It blanks the display.

There are significant differences among the storage key routines. As a protection against error by the user, information for a second observation will not be stored unless data from a first observation has already been entered. Otherwise, an error condition will be signaled.

The reference ship storage function is complicated by two factors. First, once the reference ship's course and speed have been entered, the user may desire to change only one or the other. Consequently, if the course and speed storage flag is already set, only one or the other item is required. Second, the CPA algorithm is based on all observations being made while the reference ship maintains a constant course and speed. Consequently, if the reference ship course and speed storage flag is set, entering a course or speed change will reset the storage flags for all observations on all contacts.

Finally, the first and second observation keys have a secondary function in connection with the save key described below. Consequently, if the solution flag is set, depressing one of these keys will load a CPU register with a value indicating which key has been depressed, and the normal function of the key will be suppressed.

3. The Compute Key

The following functions are performed by the compute key:

a. It verifies that only a contact identification number is presently entered in the display, and that storage flags for the reference ship's data and both observations for the indicated contact are set. Otherwise, it indicates an error and returns to the control loop.

b. It transfers contact observation data and reference ship's data to RAM register locations as required by the CPA algorithm, and executes the algorithm.

c. It sets the solution and reference ship's course and speed flags, resets all other flags except observation flags for other contacts, and restores the reference ship's data to the RAM register location required by the control program.

d. It zeroes all counters, and reinitializes the CPU registers as required by the control program.

e. Finally, it displays the contact's number and the time, range, and bearing of its CPA.

4. Output Keys

The two output keys first verify that the solution flag is set, and then display the requested information. Otherwise an error will be indicated. The CPA key displays the contact identification number and the time, range, and bearing of its CPA. The "targetship" key will display the contact's identification number and its course and speed.

5. The Save Key

After executing the CPA algorithm for a selected contact, the user has the option of saving the contact's first or second observation data for future use. The save key performs the following functions:

a. It verifies that the solution flag is set and that either the first or second observation key has been depressed . Otherwise, it indicates an error and returns to the control loop.

b. If the second observation key has been depressed, it transfers the second observation data to the solution contact's first observation RAM location, thus destroying the first observation information.

c. It resets the first observation location storage flag for the solution contact.

6. Clear Display Key and Reinitialization Button

The clear display key blanks the display and resets all flags except storage flags. This is to be distinguished from the reinitialization button, a hardware feature of the MCS-4 microcomputer set which first zeroes memory and all CPU registers, and then restarts the program.

7. The Decimal Point Key

The decimal point key verifies that the decimal point flag is not set and that the minimum number of digits required for the current input item have been entered. Otherwise, an error is indicated and the control returns to the control loop. The decimal point is then displayed in a position corresponding to the present input, and the decimal point flag is set.

8. The Number Keys

There are ten digit keys, each performing the same functions:

a. Each verifies that digits may be entered by inspecting the minimum digit counter: unless a selection key has been depressed, this counter will indicate that the maximum number of digits has already been entered. Otherwise, it indicates an error and returns.

b. It increments the maximum digit counter.

c. If the decimal point flag is set, it displays the newly selected digit to the right of the decimal point. Otherwise, it enters the new digit to the left of the decimal point, shifting the already entered digits left.

d. If the minimum number of digits has not already been entered, it increments the minimum digit counter. If the minimum number of digits has now been achieved, it sets the input flag for the item being entered.

APENDIX C

ESTIMATES OF MANUFACTURING COSTS

A. Estimated Cost Breakdown of the MCS-4 prototype

COMPONENT	QUANTITY	$ COST PER UNIT
Power Supply (+15v,-10v)	1	60
Clock	1	10
1702 PROM	13	45
4002 RAM	9	30
4004 CPU	1	60
4008 Memory Interface	1	30
4009 Memory Interface	1	30
P.C. Board	1	75
Display	1	80
Package	1	75
Total Cost		$1275

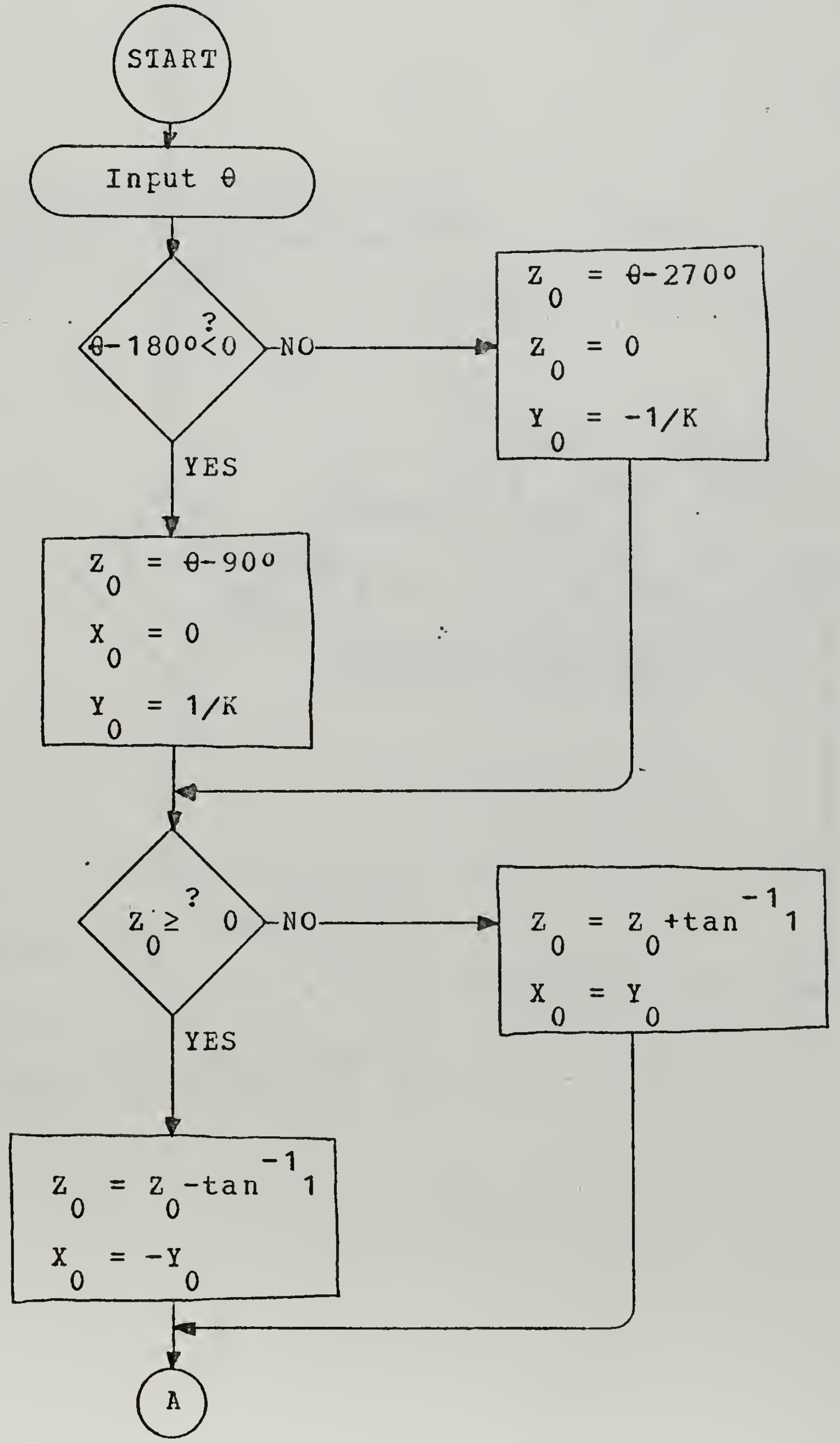
B. Estimated Cost Breakdown of Mass-Produced Computers

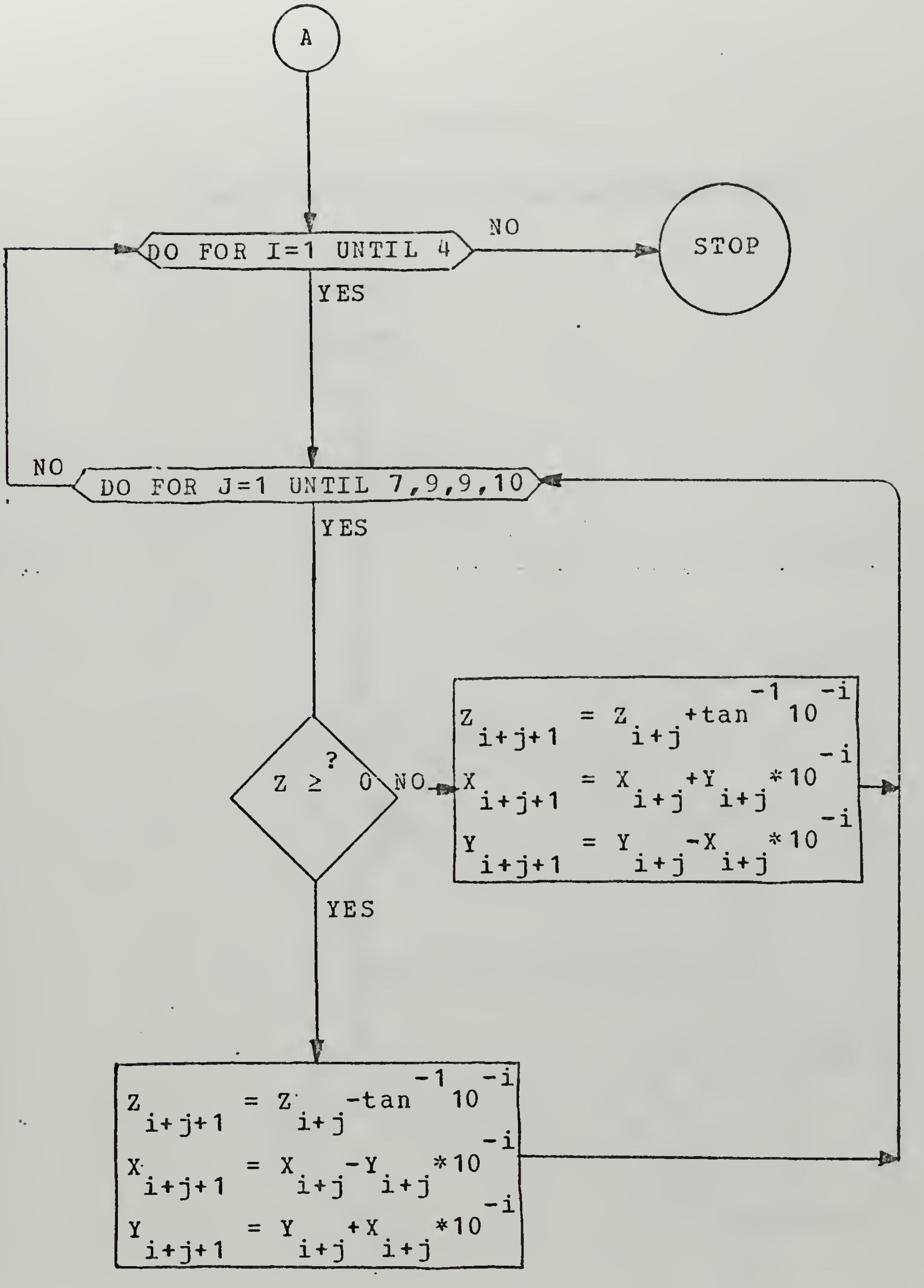
COMPONENT	$ COST PER UNIT IN QUANTITY RANGE		
	25-99	100-999	1000 PLUS
Power Supply	40	25	15
Clock	10	5	3
4001 ROM (13)	25.50	15	5
4002 RAM (9)	21	15	5
4004 CPU	42	30	20
P.C. Board	25	10	7
Display	50	30	20
Package	40	25	15
Computer Total Cost	$727.50	$455.00	$190.00

Note: A six hundred dollar initial etching cost per ROM, totaling $7800 regardless of quantity, was not included in the above cost estimates for mass production.

APPENDIX D

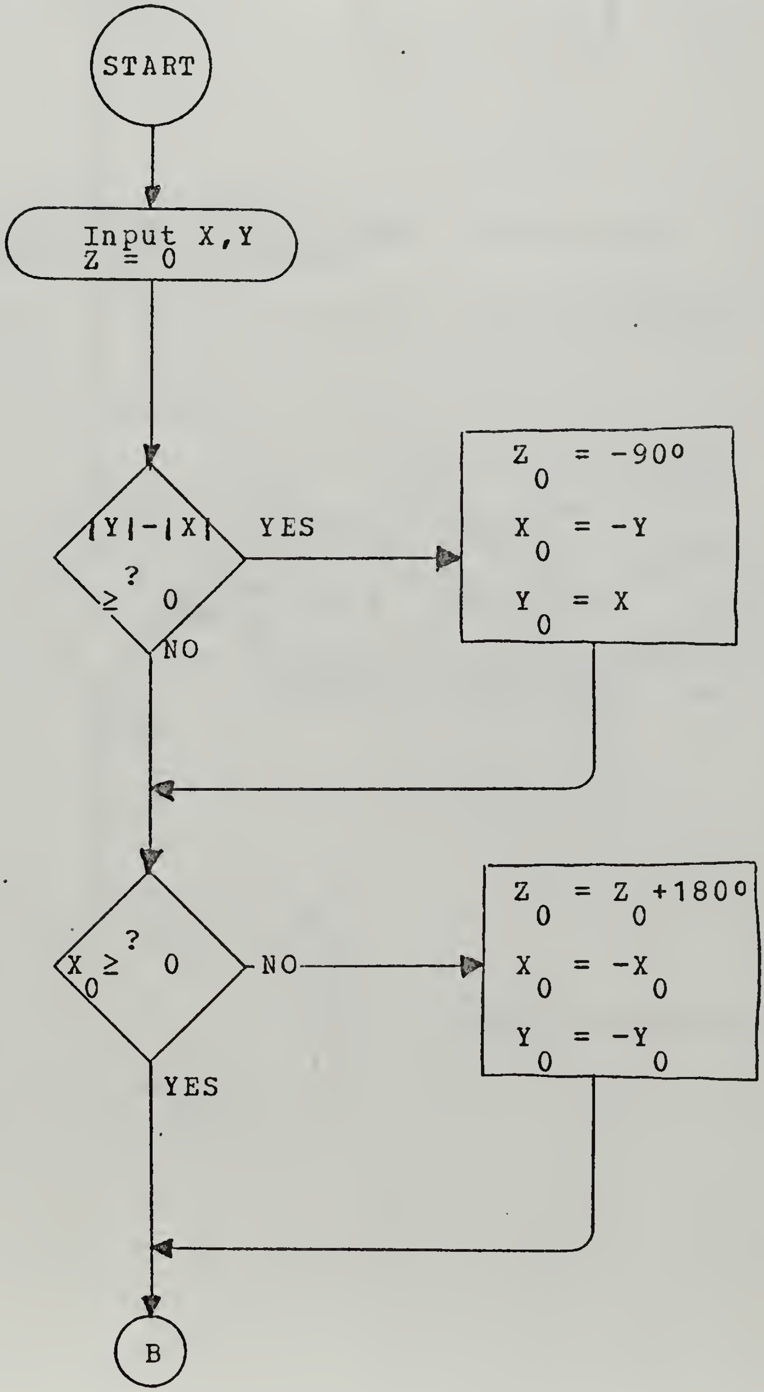
FLOW CHART FOR CORDIC ROTATION MODE

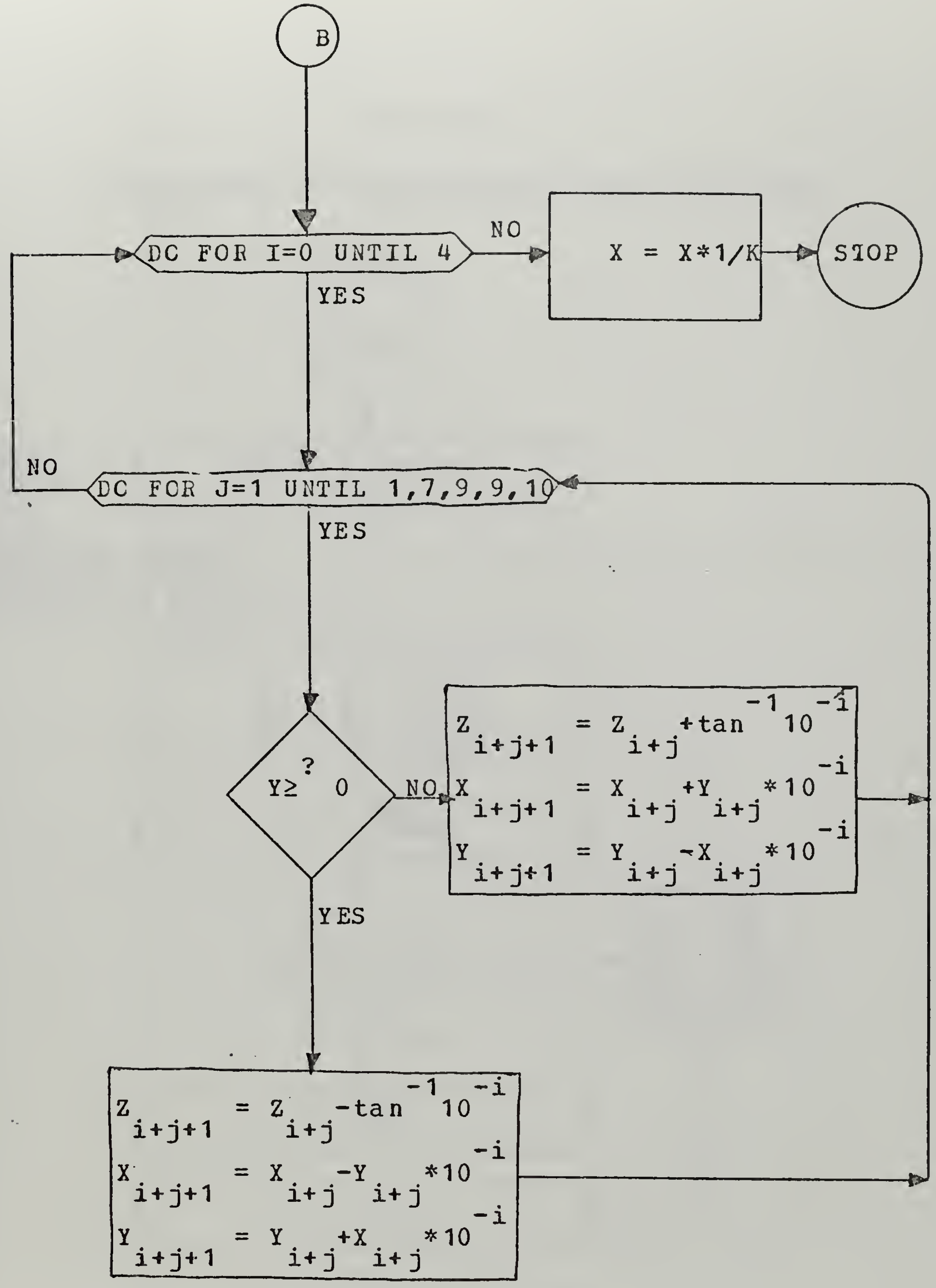



A
DO FOR I=1 UNTIL 4
NO
STOP
YES
NO
DO FOR J=1 UNTIL 7,9,9,10
YES
$Z \overset{?}{\geq} 0$
NO
$Z_{i+j+1} = Z_{i+j} + \tan^{-1} 10^{-i}$
$X_{i+j+1} = X_{i+j} + Y_{i+j} * 10^{-i}$
$Y_{i+j+1} = Y_{i+j} - X_{i+j} * 10^{-i}$
YES
$Z_{i+j+1} = Z_{i+j} - \tan^{-1} 10^{-i}$
$X_{i+j+1} = X_{i+j} - Y_{i+j} * 10^{-i}$
$Y_{i+j+1} = Y_{i+j} + X_{i+j} * 10^{-i}$

APPENDIX E

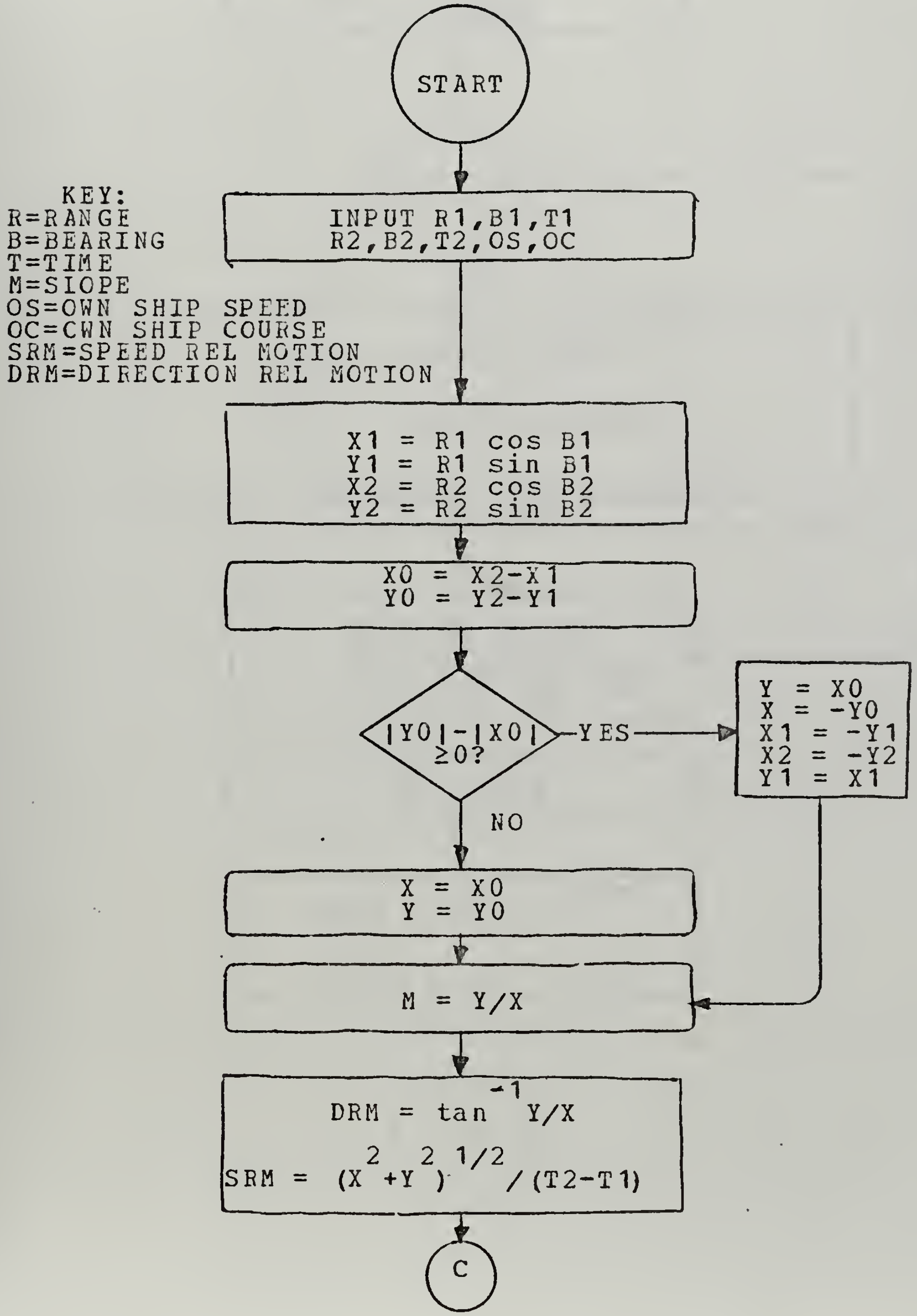
FLOW CHART FOR CORDIC VECTORING MODE

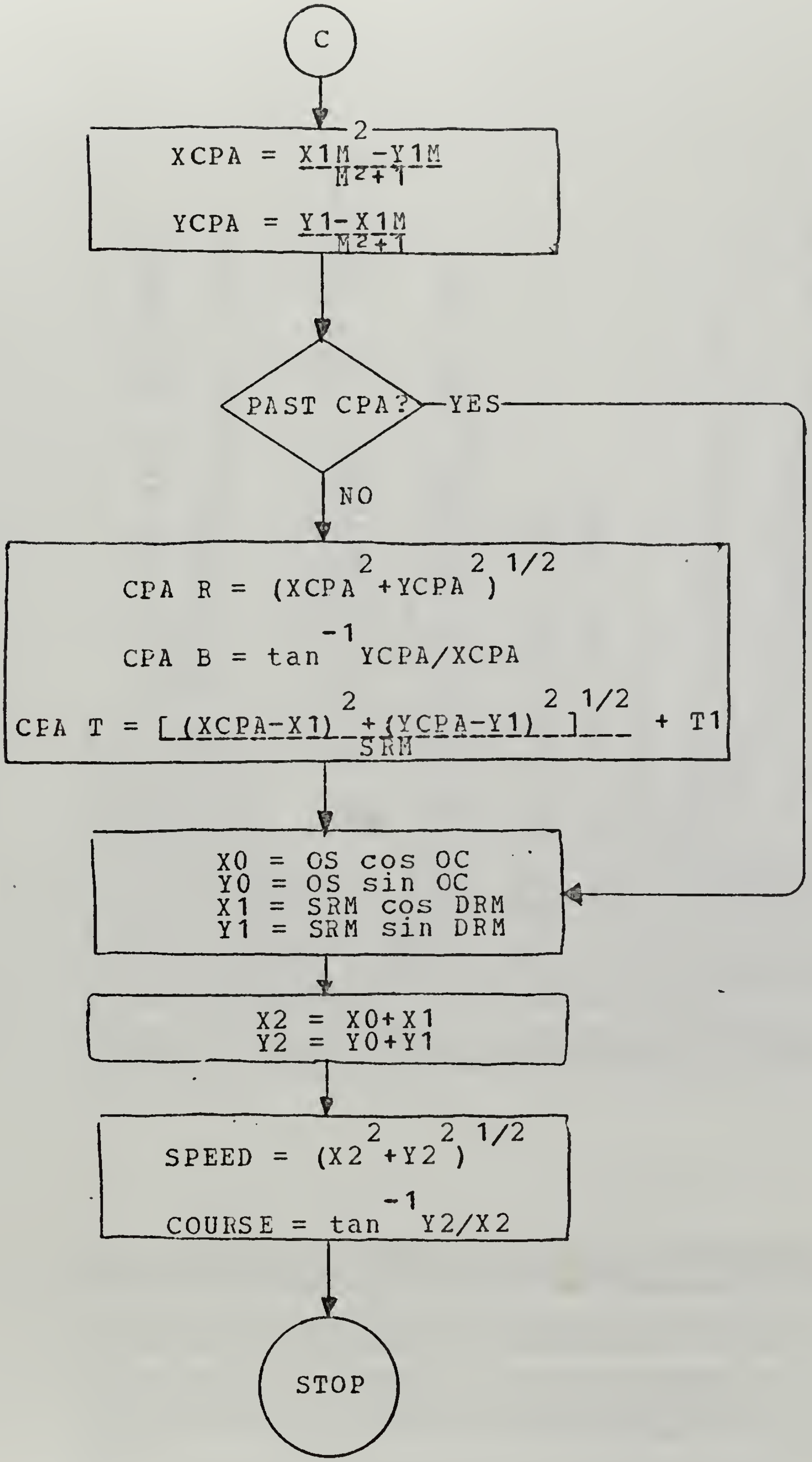



B
DC FOR I=0 UNTIL 4
NO
X = X*1/K
STOP
YES
NO
DC FOR J=1 UNTIL 1,7,9,9,10
YES
$Y \overset{?}{\geq} 0$
NO
$Z_{i+j+1} = Z_{i+j} + \tan^{-1} 10^{-i}$
$X_{i+j+1} = X_{i+j} + Y_{i+j} * 10^{-i}$
$Y_{i+j+1} = Y_{i+j} - X_{i+j} * 10^{-i}$
YES
$Z_{i+j+1} = Z_{i+j} - \tan^{-1} 10^{-i}$
$X_{i+j+1} = X_{i+j} - Y_{i+j} * 10^{-i}$
$Y_{i+j+1} = Y_{i+j} + X_{i+j} * 10^{-i}$

APPENDIX F

FLOW CHART FOR MANEUVERING BOARD ALGORITHM




C
XCPA = (X1M² - Y1M)/(M²+1)
YCPA = (Y1 - X1M)/(M²+1)
PAST CPA?
YES
NO
CPA R = (XCPA² + YCPA²)^1/2
CPA B = tan⁻¹ YCPA/XCPA
CPA T = [(XCPA-X1)² + (YCPA-Y1)²]^1/2 / SRM + T1
X0 = OS cos OC
Y0 = OS sin OC
X1 = SRM cos DRM
Y1 = SRM sin DRM
X2 = X0+X1
Y2 = Y0+Y1
SPEED = (X2² + Y2²)^1/2
COURSE = tan⁻¹ Y2/X2
STOP

PROGRAM LISTING

```
* MCS-4 ASSEMBLY *
0001 0000 00 000          REM CONTROL PROGRAM;
0002 0000 00 000          NOP;
0003 0001 00 001          REM BLANK DISPLAY;
0004 0001 00 001          FIM R0 "00"; LDM 0; XCH RB; LDM "F"; JMS DCRAM;
0005 0008 00 008          REM TURN OFF ERROR LIGHT;
0006 0008 00 008          FIM RC "40"; SRC RC; LDM 0; WRR;
0007 0013 00 00D          REM FIRST TIME, MAKE SURE KEY NOT ALREADY DOWN;
0008 0013 00 00D          JUN CHECKNOSTROBE;
0009 0015 00 00F          CHECKSTROBE:
0010 0015 00 00F          JMS DISPLAY;
0011 0017 00 011          REM WAIT FOR KEY TO BE DEPRESSED;
0012 0017 00 011          FIM RC "10"; SRC RC; RDR; RAL; JCN ZCY CHECKSTROBE;
0013 0024 00 018          REM WAIT FOR KEY TO STOP BOUNCING;
0014 0024 00 018          JMS WAIT5;
0015 0026 00 01A          REM STORE KEY'S 4-BIT DATA;
0016 0026 00 01A          FIM RA "00"; SRC RA; RDR; XCH RD;
0017 0031 00 01F          REM MAKE SURE KEY STILL DOWN, HENCE DATA GOOD;
0018 0031 00 01F          SRC RC; RDR; RAL; JCN ZCY CHECKSTROBE;
0019 0036 00 024          XCH R0;
0020 0037 00 025          REM TURN OFF ERROR LIGHT;
0021 0037 00 025          FIM RA "40"; SRC RA; LDM 0; WRR;
0022 0042 00 02A          LD R0;
0023 0043 00 02B          REM USE DATA BITS AS TABLE INDEX;
0024 0043 00 02B          FIM R0 KEYTAB;
0025 0045 00 02D          RAL; JCN ZCY NUMBERSIDE;
0026 0048 00 030          INC R0; INC RC;
0027 0050 00 032          NUMBERSIDE: LD RD;
0028 0051 00 033          CLC; ADD RD; JCN ZCY DC1; INC R0;
0029 0056 00 038          DC1: CLC; ADD R1; XCH R1; JCN ZCY DC2; INC R0;
0030 0062 00 03E          REM JUMP INTO TABLE;
0031 0062 00 03E          DC2: JIN R0;
0032 0063 00 03F          REM ALL TABLE ROUTINES RETURN HERE;
0033 0063 00 03F          CHECKNOSTROBE: JMS DISPLAY; FIM RC "10"; SRC RC;
0034 0068 00 044          REM MAKE SURE KEY HAS BEEN RELEASED;
0035 0068 00 044          RDR; RAL; JCN NZCY CHECKNOSTROBE;
0036 0072 00 048          REM IF YES, LOOP;
0037 0072 00 048          JUN CHECKSTROBE;
0038 0074 00 04A          REM THE TABLE;
```

```
0039 0074 00 04A          KEYTAB:
0040 0074 00 04A          JUN CHECKNOSTROBE;
0041 0076 00 04C          JUN NUMBER;
0042 0078 00 04E          JUN NUMBER;
0043 0080 00 050          JUN NUMBER;
0044 0082 00 052          JUN NUMBER;
0045 0084 00 054          JUN NUMBER;
0046 0086 00 056          JUN NUMBER;
0047 0088 00 058          JUN NUMBER;
0048 0090 00 05A          JUN NUMBER;
0049 0092 00 05C          JUN NUMBER;
0050 0094 00 05E          JUN CHECKNOSTROBE;
0051 0096 00 060          JUN DECIMAL;
0052 0098 00 062          JUN CHECKNOSTROBE;
0053 0100 00 064          JUN CLEARKEY;
0054 0102 00 066          JUN CHECKNOSTROBE;
0055 0104 00 068          JUN CHECKNOSTROBE;
0056 0106 00 06A          JUN NUMBER;
0057 0108 00 06C          JUN COMPUTE;
0058 0110 00 06E          JUN OWNSHIP;
0059 0112 00 070          JUN COURSE;
0060 0114 00 072          JUN SAVE;
0061 0116 00 074          JUN SPEED;
0062 0118 00 076          JUN RANGE;
0063 0120 00 078          JUN BEARING;
0064 0122 00 07A          JUN TARGETSHIP;
0065 0124 00 07C          JUN CHECKNOSTROBE;
0066 0126 00 07E          JUN CPA;
0067 0128 00 080          JUN CHECKNOSTROBE;
0068 0130 00 082          JUN SECOND;
0069 0132 00 084          JUN FIRST;
0070 0134 00 086          JUN TIME;
0071 0136 00 088          JUN TARGETNUMBER;
0072 0138 00 08A
0073 0138 00 08A          SYN BLKTO=10,REGTO=12,PREGTO=13,BLKFM=11,REGFM=6,
0074 0138 00 08A          PREGFM=7,TEMP=9;
0075 0138 00 08A
0076 0138 00 08A          REM TO TRANSFER FROM ONE RAM BLOCK TO ANOTHER;
0077 0138 00 08A          BLOCKTRANS:
0078 0138 00 08A          LD BLKFM; DCL; SRC REGFM; RDM; XCH TEMP;
0079 0143 00 08F          LD BLKTO; DCL; SRC REGTO; LD TEMP; WRM;
0080 0148 00 094          INC PREGFM; ISZ PREGTO BLOCKTRANS; BBL 0;
0081 0152 00 098
0082 0152 00 098          SYN BLKTO,REGTO,PREGTO,BLKFM,REGFM,PREGFMTEMP;
0083 0152 00 098
0084 0152 00 098          SYN CONTNR=6,TARGNR=11,BLKCODE=10,REGNR=12,PREGNR=13;
0085 0152 00 098
0086 0152 00 098          REM TO CONVERT TARGET NUMBER TO STORAGE LOCATION;
```

```
0087 0152 00 098          BLOCKSETUP:
0088 0152 00 098          CLC; LDM 7; SUB TARGNR; JCN NZCY BLK1;
0089 0157 00 09D          CMA; XCH TARGNR; LDM 2; XCH BLKCODE; JUN BLKJOIN;
0090 0163 00 0A3          BLK1: LDM 1; XCH BLKCODE;
0091 0165 00 0A5          BLKJOIN: LD TARGNR; ADD TARGNR; ADD CONTNR;
0092 0168 00 0A8          XCH REGNR; LDM 0; XCH FREGNR; BBL 0;
0093 0172 00 0AC          SYN CONTNR,TARGNR,BLKCODE,REGNR,FREGNR;
0094 0172 00 0AC
0095 0172 00 0AC          REM CHANGES F'S TO 0'S;
0096 0172 00 0AC          UNF:
0097 0172 00 0AC          SRC RC; RDM; CMA; JCN NZAC FOK; LDM 0; WRM;
0098 0179 00 0B3          FOK: ISZ RD UNF; BBL 0;
0099 0182 00 0B6
0100 0182 00 0B6          REM DETERMINES IF ENTIRE DISPLAY OR ONLY PART
0101 0182 00 0B6              MUST BE BLANKED;
0102 0182 00 0B6          TESTSOL:
0103 0182 00 0B6          FIM RC "20"; SRC RC; RDM; JCN ZAC TS1;
0104 0188 00 0BC          LDM 0; WRM;
0105 0190 00 0BE          FIM RC "00"; LDM 0; XCH RB; LDM "F"; JMS DCRAM;
0106 0197 00 0C5          LDM 0; WR0; WR1;WR2; BBL 0;
0107 0202 00 0CA          TS1: FIM RC "00"; LD RA; XCH RD; LDM "F"; JMS DCRAM;
0108 0209 00 0D1          BBL 1;
0109 0210 00 0D2
0110 0210 00 0D2          REM BLANKS DISPLAY AND INSERTS NUMBER OF
0111 0210 00 0D2              OBSERVATION TO BE SAVED;
0112 0210 00 0D2          SEESAVE:
0113 0210 00 0D2          XCH R6;
0114 0211 00 0D3          FIM RC "00"; LDM 1; XCH RB; LDM "F"; JMS DCRAM;
0115 0218 00 0DA          FIM RC "00"; LD R6; SRC RC; WRM;
0116 0223 00 0DF          LDM 0; WR0; WR1; WR2;
0117 0227 00 0E3          JUN CHECKNOSTROBE;
0118 0229 00 0E5
0119 0229 00 0E5          BEGIN 256;
0120 0256 01 100          REM RETURNS TO MAIN RAM BANK AND
0121 0256 01 100              TURNS ON ERROR LIGHT;
0122 0256 01 100          BB1:
0123 0256 01 100          LDM 0; DCL;
0124 0258 01 102          FIM RA "40"; SRC RA; LDM 15; WRR;
0125 0263 01 107          JUN CHECKNOSTROBE;
0126 0265 01 109
0127 0265 01 109          REM  FUNCTIONS PERFORMED BY KEY ROUTINES
0128 0265 01 109               DESCRIBED IN APPENDIX B.
0129 0265 01 109
0130 0265 01 109               FLAGS AS FOLLOWS:
0131 0265 01 109               10 - OWN SHIP COURSE AND SPEED STORAGE FLAG
0132 0265 01 109               11 - OBSERVATION FORMAT FLAG
0133 0265 01 109               12 - COURSE AND SPEED FORMAT FLAG
0134 0265 01 109               13 - DECIMAL POINT FLAG
```

```
0135 0265 01  109            14 - CONTACT ID NUMBER SELECTION FLAG
0136 0265 01  109            15 - RANGE SELECTION FLAG
0137 0265 01  109            16 - TIME SELECTION FLAG
0138 0265 01  109            17 - BEARING SELECTION FLAG
0139 0265 01  109            18 - COURSE SELECTION FLAG
0140 0265 01  109            19 - SPEED SELECTION FLAG
0141 0265 01  109            1A - CONTACT ID NUMBER INPUT FLAG
0142 0265 01  109            1B - RANGE INPUT FLAG
0143 0265 01  109            1C - TIME INPUT FLAG
0144 0265 01  109            1D - BEARING INPUT FLAG
0145 0265 01  109            1E - COURSE INPUT FLAG
0146 0265 01  109            1F - SPEED INPUT FLAG
0147 0265 01  109            20 - SOLUTION FLAG;
0148 0265 01  109          COMPUTE:
0149 0265 01  109          FIM RC "10"; SRC RC; RDM; JCN ZAC BB1;
0150 0271 01  10F          FIM RC "12"; SRC RC; RDM; JCN NZAC BB1;
0151 0277 01  115          FIM RC "14"; SRC RC; RDM; JCN ZAC BB1;
0152 0283 01  11B          FIM RC "1A"; SRC RC; RDM; JCN ZAC BB1;
0153 0289 01  121          LDM 13; XCH RB;
0154 0291 01  123          COMCHECK: INC RD; SRC RC; RDM; JCN NZAC BB1;
0155 0296 01  128          ISZ RB COMCHECK;
0156 0298 01  12A          LDM 0; XCH R6;
0157 0300 01  12C          FIM RC "0F"; SRC RC; RDM; XCH RB;
0158 0305 01  131          LDM "F"; WRM;
0159 0307 01  133          JMS BLOCKSETUP;
0160 0309 01  135          LD RA; DCL;
0161 0311 01  137          SRC RC; RD0; JCN ZAC BB1;
0162 0315 01  13B          INC RC; SRC RC; RD0; JCN ZAC BB1;
0163 0320 01  140          LDM 0; WR0;
0164 0322 01  142          LD RC; XCH R6; LD RD; XCH R7; LD RA; XCH RB;
0165 0328 01  148          LDM 0; XCH RA; FIM RC "EC"; JMS BLOCKTRANS;
0166 0334 01  14E          LD R6; DAC; XCH R6; SRC R6; LDM 0; WR0; FIM RC "F0";
0167 0342 01  156          JMS BLOCKTRANS;
0168 0344 01  158          FIM RA "D0"; FIM RC "EC"; JMS STATUSIN;
0169 0350 01  15E          FIM RA "D6"; FIM RC "F0"; JMS STATUSIN;
0170 0356 01  164          JUN CPACOMPUTE;
0171 0358 01  166          CPARETURN:
0172 0358 01  166          FIM RA "E0"; FIM RC "DC"; JMS STATUSOUT;
0173 0364 01  16C          FIM RA "F0"; FIM RC "D6"; JMS STATUSOUT;
0174 0370 01  172          FIM RC "10"; SRC RC; LDM 15; WRM;
0175 0375 01  177          INC RD; JMS ZERORAM;
0176 0378 01  17A          INC RC; SRC RC; LDM 15; WRM;
0177 0382 01  17E          LDM 0; XCH R2;
0178 0384 01  180          FIM R4 "E0"; FIM R6 "00"; JMS TRANSLOOP;
0179 0390 01  186          LDM 0; WR0; WR3; LDM 2; WR1; WR2;
0180 0396 01  18C          FIM RE "00";
0181 0398 01  18E          JUN CHECKNOSTROBE;
0182 04CC 01  190
```

```
0183 04C0 01 190          NUMBER:
0184 04C0 01 190          LD R7; JCN ZAC BB1;
0185 04C3 01 193          INC R7;
0186 0404 01 194          FIM RA "13"; SRC RA; RDM; JCN ZAC NORMAL;
0187 0410 01 19A          LD R9; DAC; XCH R9;
0188 0413 01 19D          SRC R8; LD RC; WRM; JUN CHECKNOSTROBE;
0189 0418 01 1A2          NORMAL: LD R5; XCH R4; LD R9; XCH R3;
0190 0422 01 1A6          SHUFFLE: SRC R2; RDM; XCH RD; WRM; INC R3;
0191 0427 01 1AB          ISZ R4 SHUFFLE;
0192 0429 01 1AD          LD R6; JCN ZAC NOSET;
0193 0432 01 1B0          INC R6; LD R6; JCN NZAC NOSET;
0194 0436 01 1B4          LDM 10; XCH RB; FIM RC "14";
0195 0440 01 1B8          SEARCHHISTORY: SRC RC; RDM; JCN NZAC SETHISTORY;
0196 0444 01 1BC          INC RD; ISZ RB SEARCHHISTORY;
0197 0447 01 1BF          SETHISTORY: CLC; LDM 6; ADD RD; XCH RD;
0198 0451 01 1C3          SRC RC; LDM "F"; WRM;
0199 0454 01 1C6          NOSET: JUN CHECKNOSTROBE;
0200 0456 01 1C8
0201 0456 01 1C8          DECIMAL:
0202 0456 01 1C8          LD R6; JCN NZAC BB1;
0203 0459 01 1CB          FIM RC "13"; SRC RC; RDM; JCN NZAC BB1;
0204 0465 01 1D1          FIM RC "16"; SRC RC; RDM; JCN ZAC DCM1;
0205 0471 01 1D7          SRC RE; LDM 2; WR2; JUN DCMJOIN;
0206 0476 01 1DC          DCM1: INC RD; SRC RC; RDM; JCN ZAC DCM2;
0207 0481 01 1E1          SRC RE; LDM 2; WR1; JUN DCMJOIN;
0208 0486 01 1E6          DCM2: INC RD; SRC RC; RDM; JCN ZAC DCM3;
0209 0491 01 1EB          SRC RE; LDM 2; WR1; JUN DCMJOIN;
0210 0496 01 1F0          DCM3: INC RD; SRC RC; RDM; JCN ZAC BB1;
0211 0501 01 1F5          SRC RE; LDM 8; WR0;
0212 0504 01 1F8          DCMJOIN: FIM RC "13"; SRC RC; LDM 15; WRM; XCH R7;
0213 0510 01 1FE          JUN CHECKNOSTROBE;
0214 0512 02 200
0215 0512 02 200          BEGIN 512;
0216 0512 02 200          BB2:
0217 0512 02 200          LDM 0; DCL;
0218 0514 02 202          FIM RA "40"; SRC RA; LDM 15; WRR;
0219 0519 02 207          JUN CHECKNOSTROBE;
0220 0521 02 209
0221 0521 02 209          OWNSHIP:
0222 0521 02 209          FIM RC "10"; SRC RC; RDM; JCN ZAC NEEDBOTH;
0223 0527 02 20F          FIM RC "1E"; SRC RC; RDM; XCH R4; LD R4;
0224 0533 02 215          JCN ZAC BOTHBLANK;
0225 0535 02 217          FIM RA "06"; FIM RC "D6"; LDM 12; XCH RF;
0226 0541 02 21D          JMS TRANSPART;
0227 0543 02 21F          BOTHBLANK: FIM RC "1F"; SRC RC; LD R4; CLC; ADM;
0228 0549 02 225          JCN ZAC BB2;
0229 0551 02 227          RDM; XCH R4;
0230 0553 02 229          LDM 1; DCL; FIM RC "0C"; LDM 0;
```

```
0231 0558 02 22E          VITIATE1: SRC RC; WR0; ISZ RC VITIATE1;
0232 0562 02 232          LDM 2; DCL; LDM 0;
0233 0565 02 235          VITIATE2: SRC RC; WR0; INC RC; ISZ RD VITIATE2;
0234 0570 02 23A          LDM 0; DCL;
0235 0572 02 23C          LD R4; JCN ZAC JOINOS;
0236 0575 02 23F          FIM RA "00"; FIM RC "D0"; LDM 13; XCH RF;
0237 0581 02 245          JMS TRANSPART;
0238 0583 02 247          JUN JOINCS;
0239 0585 02 249          NEEDBOTH: FIM RC "1E"; SRC RC; RDM; JCN ZAC BB2;
0240 0591 02 24F          INC RD;
0241 0592 02 250          SRC RC; RDM; JCN ZAC BB2;
0242 0596 02 254          FIM R4 "00"; FIM R6"DC"; JMS TRANSLOOP;
0243 0602 02 25A          JOINCS: FIM RC "10"; SRC RC; LDM "F"; WRM;
0244 0607 02 25F          INC RD; JMS ZERORAM;
0245 0610 02 262          FIM RC "D0"; JMS UNF; FIM R6 "00";
0246 0616 02 268          FIM RC "00"; LDM 0; XCH RB; LDM "F"; JMS DORAM;
0247 0623 02 26F          LDM 0; WR0; WR1;
0248 0626 02 272          JUN CHECKNOSTROBE;
0249 0628 02 274
0250 0628 02 274          COURSE:
0251 0628 02 274          FIM RC "11"; SRC RC; RDM; JCN NZAC BB2;
0252 0634 02 27A          LD R6; JCN ZAC COURSEOK;
0253 0637 02 27D          FIM RC "18"; SRC RC; RDM; JCN ZAC BB2;
0254 0643 02 283          COURSEOK: JMS UNSET7;
0255 0645 02 285          FIM RC "12"; SRC RC; LDM "F"; WRM;
0256 0650 02 28A          FIM RC "18"; SRC RC; WRM;
0257 0654 02 28E          FIM RC "1E"; SRC RC; LDM 0; WRM;
0258 0659 02 293          FIM RA "6C"; JMS TESTSCL;
0259 0663 02 297          LDM 0; WR1; FIM R6 "FD"; FIM R4 "0D";
0260 0669 02 29D          FIM R8 "07"; JUN CHECKNOSTROBE;
0261 0673 02 2A1
0262 0673 02 2A1          FIRST:
0263 0673 02 2A1          FIM RC "20"; SRC RC; RDM; JCN ZAC SETCHECK1;
0264 0679 02 2A7          LDM 1; JUN SEESAVE;
0265 0682 02 2AA          SETCHECK1: JMS SAME12;
0266 0684 02 2AC          JMS BLOCKSETUP;
0267 0686 02 2AE          SAMEFS:
0268 0686 02 2AE          LDM 0; XCH RB;
0269 0688 02 2B0          FIM R6 "00"; JMS BLOCKTRANS;
0270 0692 02 2B4          JMS UNF; LDM 15; WR0;
0271 0696 02 2B8          LDM 0; DCL;
0272 0698 02 2BA          FIM RC "11"; JMS ZERORAM;
0273 0702 02 2BE          FIM RC "00"; LDM 0; XCH RB; LDM "F"; JMS DORAM;
0274 0709 02 2C5          LDM 0; WR1; WR2;
0275 0712 02 2C8          FIM R6 "00"; JUN CHECKNOSTROBE;
0276 0716 02 2CC
0277 0716 02 2CC          SECOND:
0278 0716 02 2CC          FIM RC "20"; SRC RC; RDM ; JCN ZAC SETCHECK2;
```

```
0279 0722 02 2D2        LDM 2; JUN SEESAVE;
0280 0725 02 2D5        SETCHECK2: JMS SAME12;
0281 0727 02 2D7        JMS BLOCKSETUP;
0282 0729 02 2D9        LD RA; DCL;
0283 0731 02 2DB        SRC RC; RD0; JCN ZAC BB2;
0284 0735 02 2DF        INC RC;
0285 0736 02 2E0        JUN SAMEFS;
0286 0738 02 2E2
0287 0738 02 2E2        SAME12:
0288 0738 02 2E2        LDM 12; XCH RB; FIM RC "1A";
0289 0742 02 2E6        ALLIN: SRC RC; RDM; JCN ZAC BB2;
0290 0746 02 2EA        INC RD; ISZ RB ALLIN;
0291 0749 02 2ED        LDM 0; XCH R6; FIM RC "CF"; SRC RC; RDM;
0292 0755 02 2F3        XCH RB; BBL 0;
0293 0757 02 2F5
0294 0757 02 2F5        BEGIN 768;
0295 0768 03 300        BB3:
0296 0768 03 300        FIM RA "40"; SRC RA; LDM 15; WRR;
0297 0773 03 305        JUN CHECKNOSTROBE;
0298 0775 03 307
0299 0775 03 307        TARGETNUMBER:
0300 0775 03 307        FIM RC "12"; SRC RC; RDM; JCN NZAC BB3;
0301 0781 03 30D        LD R6; JCN ZAC NOK;
0302 0784 03 310        FIM RC "14"; SRC RC; RDM; JCN ZAC BB3;
0303 0790 03 316        NOK: JMS UNSET7;
0304 0792 03 318        FIM RC "11"; SRC RC; LDM "F"; WRM;
0305 0797 03 31D        FIM RC "14"; SRC RC; WRM;
0306 0801 03 321        FIM RC "1A"; SRC RC; LDM "0"; WRM;
0307 0806 03 326        FIM RA "FF"; JMS TESTSCL;
0308 0810 03 32A        FIM R6 "FF"; FIM R4 "0F";
0309 0814 03 32E        FIM R8 "0F";
0310 0816 03 330        JUN CHECKNOSTROBE;
0311 0818 03 332
0312 0818 03 332        RANGE:
0313 0818 03 332        FIM RC "12"; SRC RC; RDM; JCN NZAC BB3;
0314 0824 03 338        LD R6; JCN ZAC RANGEOK;
0315 0827 03 33B        FIM RC "15"; SRC RC; RDM; JCN ZAC BB3;
0316 0833 03 341        RANGEOK: JMS UNSET7;
0317 0835 03 343        FIM RC "11"; SRC RC; LDM "F"; WRM;
0318 0840 03 348        FIM RC "15"; SRC RC; WRM;
0319 0844 03 34C        FIM RC "1B"; SRC RC; LDM 0; WRM;
0320 0849 03 351        FIM RA "0A"; JMS TESTSCL;
0321 0853 03 355        FIM R6 "FA"; FIM R4 "0A";
0322 0857 03 359        FIM R8 "00"; JUN CHECKNOSTROBE;
0323 0861 03 35D
0324 0861 03 35D        BEARING:
0325 0861 03 35D        FIM RC "12"; SRC RC; RDM; JCN NZAC BB3;
0326 0867 03 363        LD R6; JCN ZAC BEARINGOK;
```

```
0327 0870 03 366          FIM RC "17"; SRC RC; RDM; JCN ZAC BB3;
0328 0876 03 36C          BEARINGOK:
0329 0876 03 36C          JMS UNSET7;
0330 0878 03 36E          FIM RC "11"; SRC RC; LDM "F"; WRM;
0331 0883 03 373          FIM RC "17"; SRC RC; WRM;
0332 0887 03 377          FIM RC "1D"; SRC RC; LDM 0; WRM;
0333 0892 03 37C          FIM RA "6C"; JMS TESTSCL;
0334 0896 03 380          LDM 0; WR1;
0335 0898 03 382          FIM R6 "FD"; FIM R4 "0D";
0336 0902 03 386          FIM R8 "07";
0337 0904 03 388          JUN CHECKNOSTROBE;
0338 0906 03 38A
0339 0906 03 38A          TIME:
0340 0906 03 38A          FIM RC "12"; SRC RC; RDM; JCN NZAC BB3;
0341 0912 03 390          LD R6; JCN ZAC TIMEOKOK;
0342 0915 03 393          FIM RC "16"; SRC RC; RDM; JCN ZAC BB3;
0343 0921 03 399          TIMEOKOK: JMS UNSET7;
0344 0923 03 39B          FIM RC "11"; SRC RC; LDM "F"; WRM;
0345 0928 03 3A0          FIM RC "16"; SRC RC; WRM;
0346 0932 03 3A4          FIM RC "1C"; SRC RC; LDM 0; WRM;
0347 0937 03 3A9          FIM RA "AB"; JMS TESTSCL;
0348 0941 03 3AD          LDM 0; WR2; FIM R6 "FC"; FIM R4 "0C";
0349 0947 03 3B3          FIM R8 "0B"; JUN CHECKNOSTROBE;
0350 0951 03 3B7
0351 0951 03 3B7          SPEED:
0352 0951 03 3B7          FIM RC "11"; SRC RC; RDM; JCN NZAC BB3;
0353 0957 03 3BD          LD R6; JCN ZAC SPEEDOK;
0354 0960 03 3C0          FIM RC "19"; SRC RC; RDM; JCN ZAC BB3;
0355 0966 03 3C6          SPEEDOK: JMS UNSET7;
0356 0968 03 3C8          FIM RC "12"; SRC RC; LDM "F"; WRM;
0357 0973 03 3CD          FIM RC "19"; SRC RC; WRM;
0358 0977 03 3D1          FIM RC "1F"; SRC RC; LDM 0; WRM;
0359 0982 03 3D6          FIM RA "0D"; JMS TESTSCL;
0360 0986 03 3DA          LDM 0; WR0; FIM R6 "FE"; FIM R4 "0E";
0361 0992 03 3E0          FIM R8 "01"; JUN CHECKNOSTROBE;
0362 0996 03 3E4
0363 0996 03 3E4          REM ZEROES FLAGS 13 THRU 19;
0364 0996 03 3E4          UNSET7:
0365 0996 03 3E4          FIM RC "13"; LDM 9; XCH R8; LDM 0;
0366 1001 03 3E9          JMS DORAM; BBL 0;
0367 1004 03 3EC
0368 1004 03 3EC          BEGIN 1024;
0369 1024 04 400          BB4:
0370 1024 04 400          FIM RA "40"; SRC RA; LDM 15; WRR;
0371 1029 04 405          JUN CHECKNOSTROBE;
0372 1031 04 407
0373 1031 04 407          SAVE:
0374 1031 04 407          FIM RC "20"; SRC RC; RDM; JCN ZAC BB4;
```

```
0375 1037 04 40D          LD R6; JCN ZAC BB4;
0376 1040 04 410          DAC; XCH R6;
0377 1042 04 412          FIM RC "0F"; SRC RC; RDM; XCH RB;
0378 1047 04 417          JMS BLOCKSETUP;
0379 1049 04 419          LD RA; DCL;
0380 1051 04 41B          LD R6; JCN NZAC SAVEIT;
0381 1054 04 41E          SRC RC; LDM "F"; WR0;
0382 1057 04 421          JUN SAVEND:
0383 1059 04 423          SAVEIT: LD RC; XCH R4; LD RC; DAC; XCH R6;
0384 1064 04 428          LDM 0; XCH RD; LDM 0; XCH R7; JMS TRANSLOOP;
0385 1070 04 42E          LDM 0; XCH R6;
0386 1072 04 430          SAVEND: LDM "F"; WR0;
0387 1074 04 432          LDM 0; DCL; FIM RC "00"; SRC RC; LDM "F"; WRM;
0388 1081 04 439          JUN CHECKNOSTRCBE;
0389 1083 04 43B
0390 1083 04 43B          CPA:
0391 1083 04 43B          FIM RC "20"; SRC RC; RDM; JCN ZAC BB4;
0392 1089 04 441          FIM R4 "E0"; FIM R6 "00"; JMS TRANSLOOP;
0393 1095 04 447          LDM 0; WR0; LDM 2; WR1; WR2;
0394 1100 04 44C          JUN CHECKNOSTRCBE;
0395 1102 04 44E
0396 1102 04 44E          TARGETSHIP:
0397 1102 04 44E          FIM RC "20"; SRC RC; RDM; JCN ZAC BB4;
0398 1108 04 454          FIM R4 "F0"; FIM R6 "00"; JMS TRANSLOOP;
0399 1114 04 45A          LDM 8; WR0; LDM 2; WR1; LDM 0; WR2;
0400 1120 04 460          JUN CHECKNOSTRCBE;
0401 1122 04 462
0402 1122 04 462          REM TRANSFERS COURSE AND SPEED DATA TO
0403 1122 04 462              RAM REGISTER E ANF F STATUS CHARACTERS;
0404 1122 04 462          STATUSIN: SRC RA; RDM; SRC RC; WR0; INC RB;
0405 1127 04 467          SRC RA; RDM; SRC RC; WR1; INC RB;
0406 1132 04 46C          SRC RA; RDM; SRC RC; WR2; INC RB;
0407 1137 04 471          SRC RA; RDM; SRC RC; WR3; BBL 0;
0408 1142 04 476
0409 1142 04 476          REM TRANSFERS COURSE AND SPEED DATA OUT OF
0410 1142 04 476              STATUS CHARACTERS INTO WORKING REGISTERS;
0411 1142 04 476          STATUSOUT:
0412 1142 04 476          SRC RA; RD0; SRC RC; WRM; INC RD;
0413 1147 04 47B          SRC RA; RD1; SRC RC; WRM; INC RD;
0414 1152 04 480          SRC RA; RD2; SRC RC; WRM; INC RD;
0415 1157 04 485          SRC RA; RD3; SRC RC; WRM; BBL 0;
0416 1162 04 48A
0417 1162 04 48A          REM LOOPS APPROX. 5 MILLISECS;
0418 1162 04 48A          WAIT5:
0419 1162 04 48A          LDM 0; XCH RA; LDM 0; XCH RB;
0420 1166 04 48E          WAIT5LOOP: ISZ RB WAIT5LOOP; ISZ RA WAIT5LOOP;
0421 1170 04 492          BBL 0;
0422 1171 04 493
```

```
0423 1171 04 493          SYN P1=0,T1=2,D0=1,D1=3,D2=4,D3=11,
0424 1171 04 493          BCD1=8,CNT1=10,L1=12,L2=13,RAM1=14,RAM2=15;
0425 1171 04 493
0426 1171 04 493          REM
0427 1171 04 493          THE DISPLAY CONSISTS OF 16 LED'S AND A 16-BIT SHIFT
0428 1171 04 493          REGISTER.  A ROM PORT IS USED TO SEND THE BCD
0429 1171 04 493          INFORMATION TO THE 7447 DECODERS FOR ALL 16 LED'S
0430 1171 04 493          AT THE SAME TIME.  THE SHIFT REGISTER IS USED TO
0431 1171 04 493          ENABLE ONLY ONE OF THESE LED'S AT A PARTICULAR
0432 1171 04 493          TIME.  ROM PORT 0 IS USED FOR THE DISPLAY CHARACTER.
0433 1171 04 493          ROM PORT 1 IS USED FOR CONTROL: THE HIGH ORDER BIT
0434 1171 04 493          IS USED FOR THE DECIMAL POINT (1 MEANS OFF), THE
0435 1171 04 493          NEXT BIT IS USED TO CLOCK THE SHIFT REGISTER
0436 1171 04 493          (A CHANGE FROM 1 TO 0 CAUSES THE REGISTER TO SHIFT),
0437 1171 04 493          AND THE LOW ORDER BIT IS THE SHIFT REGISTER DATA
0438 1171 04 493          BIT.  A 0 IN THE SHIFT REGISTER ENABLES AN LED,
0439 1171 04 493          A 1 DISABLES.  THE ROUTINE IS WRIITEN SUCH THAT
0440 1171 04 493          A SINGLE 0 IS SHIFTING ACROSS THE REGISTER,
0441 1171 04 493          ENABLING EACH LED IN TURN;
0442 1171 04 493          DISPLAY:
0443 1171 04 493          REM INITIALIZE;
0444 1171 04 493          LDM 0; XCH BCD1; REM BCD PORT;
0445 1173 04 495          LDM 1; XCH CNT1; REM CONTROL PORT;
0446 1175 04 497          LDM 0; XCH P1; FIM L1 "CC";
0447 1179 04 49B          REM LOOP COUNTERS;
0448 1179 04 49B          SRC RAM1; RD0; XCH D0;
0449 1182 04 49E          RD1; XCH D1; RD2; XCH D2; RD3; XCH D3;
0450 1188 04 4A4          REM CLOCK SHIFT REGISTER;
0451 1188 04 4A4          CLOK: CLB;
0452 1189 04 4A5          LDM 4; ADD P1; SRC CNT1; WRR;
0453 1193 04 4A9          REM CLOCK NOW HIGH;
0454 1193 04 4A9          REM BLANK DISPLAY DURING SHIFT;
0455 1193 04 4A9          LDM 15; SRC BCD1; WRR;
0456 1196 04 4AC          REM HANDLE DECIMAL POINT;
0457 1196 04 4AC          LD P1; RAL; STC; RAR; XCH P1;
0458 1201 04 4B1          LD D0; RAL; XCH D0; JCN ZCY DCLOK;
0459 1206 04 4B6          LD P1; RAL; CLC; RAR; XCH P1;
0460 1211 04 4BB          REM DROP CLOCK;
0461 1211 04 4BB          DCLOK:
0462 1211 04 4BB          SRC CNT1; LD P1; WRR;
0463 1214 04 4BE          REM SEND RAM DIGIT;
0464 1214 04 4BE          SRC RAM1; RDM; SRC BCD1; WRR; INC RAM2;
0465 1219 04 4C3          LD P1; RAR; STC; RAL; XCH P1;
0466 1224 04 4C8          REM WAIT A WHILE;
0467 1224 04 4C8          LDM 0; XCH T1;
0468 1226 04 4CA          WAIT: ISZ T1 WAIT;
0469 1228 04 4CC          REM LOOP BACK TO DISPLAY REMAINING DIGITS;
0470 1228 04 4CC          ISZ L1 CLOK; LD D1; XCH D0;
```

```
0471 1232 04 4D0          LD D2; XCH D1; LD D3; XCH D2;
0472 1236 04 4D4          LDM 12; XCH L1; ISZ L2 CLOCK;
0473 1240 04 4D8          SRC CNT1; LDM 13; WRR; LDM 9; WRR;
0474 1245 04 4DD          BBL 0;
0475 1246 04 4DE          SYN P1,T1,D0,D1,D2,D3,BCD1,CNT1,L1,L2,RAM1,RAM2;
0476 1246 04 4DE
0477 1246 04 4DE          CLEARKEY:
0478 1246 04 4DE          FIM RC "00"; LDM 0; XCH RB; LDM "F"; JMS DCRAM;
0479 1253 04 4E5          LDM 0; WR0; WR1; WR2;
0480 1257 04 4E9          FIM RC "11"; JMS ZERORAM;
0481 1261 04 4ED          FIM R6 "00";
0482 1263 04 4EF          JUN CHECKNOSTROBE;
0483 1265 04 4F1          REM END CONTROL PROGRAM;
0484 1265 04 4F1
0485 1265 04 4F1          BEGIN 1280;
0486 1280 05 500          REMMANEUVERING BOARD ALGORITHM;
0487 1280 05 500
0488 1280 05 500          REM INPUT TIMES RANGES BEARINGS. CONVERT TO
0489 1280 05 500              RECTANGULAR COORDS;
0490 1280 05 500          CPACOMPUTE:
0491 1280 05 500          FIM RA "F6"; FIM RC "00"; JMS LOADBRG;
0492 1286 05 506          JMS CORDIC;
0493 1288 05 508          FIM RA "F0"; FIM RC "30"; JMS LOADRNG;
0494 1294 05 50E          FIM R2 "30"; FIM R4 "10"; FIM R6 "80"; JMS MLRT;
0495 1302 05 516          FIM R2 "30"; FIM R4 "20"; FIM R6 "90"; JMS MLRT;
0496 1310 05 51E          FIM RA "E6"; SRC RA; LDM 0; WR3; FIM RC "00";
0497 1317 05 525          JMS LOADBRG;
0498 1319 05 527          JMS CORDIC;
0499 1321 05 529          FIM RA "E0"; FIM RC "30"; JMS LOADRNG;
0500 1327 05 52F          FIM R2 "30"; FIM R4 "10"; FIM R6 "60"; JMS MLRT;
0501 1335 05 537          FIM R2 "30"; FIM R4 "20"; FIM R6 "70"; JMS MLRT;
0502 1343 05 53F          FIM R4 "60"; FIM R6 "10"; JMS TRANSLOOP;
0503 1349 05 545          FIM R4 "70"; FIM R6 "20"; JMS TRANSLOOP;
0504 1355 05 54B
0505 1355 05 54B          REM COMPUTE X,Y COORDINATES OF RELATIVE
0506 1355 05 54B              MOTION LINE;
0507 1355 05 54B          FIM R0 "10"; FIM R2 "80"; JMS SBRT;
0508 1361 05 551          FIM R0 "20"; FIM R2 "90"; JMS SBRT;
0509 1367 05 557
0510 1367 05 557          REM IF Y COORD > X COORD ROTATE POSITS +90
0511 1367 05 557              TO AVOID POSSIBLE INFINITE SLOPE OF
0512 1367 05 557              RELATIVE MOTION LINE;
0513 1367 05 557          FIM R4 "20"; FIM R6 "A0"; JMS TRANSLOOP;
0514 1373 05 55D          RD0; RAR; JCN ZCY POSY;
0515 1377 05 561          FIM R0 "A0"; JMS CPLRT;
0516 1381 05 565          POSY: FIM R4 "10"; FIM R6 "B0"; JMS TRANSLOOP;
0517 1387 05 56B          RD0; RAR; JCN ZCY POSX;
0518 1391 05 56F          FIM R0 "B0"; JMS CPLRT;
```

```
0519 1395 05 573      POSX: FIM R0 "A0"; FIM R2 "B0"; JMS SBRT;
0520 1401 05 579      RDO; RAR; JCN NZCY NOROTATE;
0521 1405 05 57D      JMS REVOLVE;
0522 1407 05 57F
0523 1407 05 57F      REM COMPUTE SLOPE OF RELATIVE MOTION LINE;
0524 1407 05 57F      NOROTATE: FIM R2 "20"; FIM R6 "10"; FIM R8 "70";
0525 1413 05 585      FIM R4 "A0"; JMS DVRT;
0526 1417 05 589
0527 1417 05 589      REM COMPUTE DRM AND SRM;
0528 1417 05 589      JMS CORDICAT;
0529 1419 05 58B      FIM RA "EA"; FIM RC "3C"; JMS LCADTIMELO;
0530 1425 05 591      FIM RA "FA"; FIM RC "4C"; JMS LCADTIMELO;
0531 1431 05 597      FIM R4 "50"; FIM R0 HRSFERMIN; JMS FERT;
0532 1437 05 59D      FIM R2 "30"; FIM R4 "5C"; FIM R6 "A0"; JMS MLRT;
0533 1445 05 5A5      FIM R2 "40"; FIM R4 "5C"; FIM R6 "B0"; JMS MLRT;
0534 1453 05 5AD      FIM RA "ED"; FIM RC "AC"; JMS LCADTIMEHI;
0535 1459 05 5B3      FIM RA "FD"; FIM RC "BC"; JMS LCADTIMEHI;
0536 1465 05 5B9      FIM R0 "A0"; FIM R2 "BC"; JMS SBRT;
0537 1471 05 5BF      RDO; RAR; JCN ZCY TIMECK;
0538 1475 05 5C3      FIM R4 "C0"; FIM R0 DAY; JMS FERT;
0539 1481 05 5C9      FIM R0 "A0"; FIM R2 "C0"; JMS ADRT;
0540 1487 05 5CF
0541 1487 05 5CF      REM COMPUTE CPA POINT IN RECTANGULAR
0542 1487 05 5CF          COORDINATES;
0543 1487 05 5CF      TIMECK: FIM R2 "20"; FIM R6 "A0"; FIM R8 "C0";
0544 1493 05 5D5      FIM R4 "D0"; JMS DVRT;
0545 1497 05 5D9      FIM R4 "70"; FIM R6 "50"; JMS TRANSLOOP;
0546 1503 05 5DF      FIM R2 "70"; FIM R4 "5C"; FIM R6 "D0"; JMS MLRT;
0547 1511 05 5E7      FIM R2 "80"; FIM R4 "DC"; FIM R6 "50"; JMS MLRT;
0548 1519 05 5EF      LDM 9; XCH R5; SRC R4; LDM 1; WRM;
0549 1524 05 5F4      FIM R2 "90"; FIM R4 "7C"; FIM R6 "30"; JMS MLRT;
0550 1532 05 5FC      FIM R0 "50"; FIM R2 "30"; JMS SBRT;
0551 1538 06 602      FIM R2 "80"; FIM R4 "7C"; FIM R6 "40"; JMS MLRT;
0552 1546 06 60A      FIM R4 "90"; FIM R6 "30"; JMS TRANSLOOP;
0553 1552 06 610      FIM R0 "30"; FIM R2 "4C"; JMS SBRT;
0554 1558 06 616      FIM R2 "50"; FIM R6 "DC"; FIM R8 "10";
0555 1564 06 61C      FIM R4 "40"; JMS DVRT;
0556 1568 06 620      FIM R2 "30"; FIM R6 "D0"; FIM R8 "20";
0557 1574 06 626      FIM R4 "40"; JMS DVRT;
0558 1578 06 62A      FIM R4 "60"; FIM R6 "A0"; JMS TRANSLOOP;
0559 1584 06 630      FIM R0 "A0"; FIM R2 "8C"; JMS SBRT;
0560 1590 06 636      RDO; XCH RE;
0561 1592 06 638
0562 1592 06 638      REM TEST FOR FAST-CPA;
0563 1592 06 638      FIM R4 "10"; FIM R6 "A0"; JMS TRANSLOOP;
0564 1598 06 63E      FIM R2 "60"; JMS SBRT;
0565 1602 06 642      NOP;
0566 1603 06 643
```

```
0567 1603 06 643        RDO; CLC; ADD RE; RAR; JCN ZCY FINDCPA;
0568 1609 06 649
0569 1609 06 649        REM PAST-CPA INDICATED BY ALL F'S OUTPUT;
0570 1609 06 649        LDM "F"; FIM RC "E0"; FLOOP: SRC R0; WRM;
0571 1614 06 64E        ISZ R1 FLOOP;
0572 1616 06 650        FIM R4 "00"; FIM R6 "AC"; JMS TRANSLOOP;
0573 1622 06 656
0574 1622 06 656        REM IF PAST-CPA COMPUTE TARGET
0575 1622 06 656             COURSE AND SPEED;
0576 1622 06 656        JUN FINDCANDS;
0577 1624 06 658
0578 1624 06 658        REM ELSE COMPUTE CPA BEARING,RANGE AND TIME;
0579 1624 06 658        FINDCPA: FIM R4 "10"; FIM R6 "60"; JMS TRANSLOOP;
0580 1630 06 65E        FIM R4 "20"; FIM R6 "7C"; JMS TRANSLOOP;
0581 1636 06 664        FIM R4 "00"; FIM R6 "AC"; JMS TRANSLOOP;
0582 1642 06 66A        JMS CORDICAT;
0583 1644 06 66C
0584 1644 06 66C        REM
0585 1644 06 66C        IF POSITIONS ROTATED,ROTATE CPA BEARING
0586 1644 06 66C             MINUS 90 DEGREES;
0587 1644 06 66C        FIM RA "E0"; SRC RA; RD3; JCN ZAC NOBACKROTATE1;
0588 1650 06 672        FIM R4 "30"; FIM R0 MINPI2; JMS FERT;
0589 1656 06 678        FIM R0 "00"; FIM R2 "3C"; JMS ADRT;
0590 1662 06 67E        SRC R0;RD0;RAR;JCN ZCY NOBACKROTATE1;
0591 1667 06 683        FIM R4 "30";FIM R0 TWOPI;JMS FERT;
0592 1673 06 689        FIM R0 "00";FIM R2 "3C";JMS ADRT;
0593 1679 06 68F
0594 1679 06 68F        REM PUT VALUES INTO OUTPUT FOR CPA BEARING
0595 1679 06 68F             AND RANGE;
0596 1679 06 68F        NOBACKROTATE1:
0597 1679 06 68F        FIM RA "08"; FIM RC "E6";
0598 1683 06 693        LDM 12;XCH RF;JMS TRANSPART;
0599 1687 06 697        FIM RA "29"; FIM RC "EC"; LDM 10; XCH RF;
0600 1693 06 69D        JMS TRANSPART;
0601 1695 06 69F        FIM RC "E5"; LDM 11; XCH RB; JMS FBACK;
0602 1701 06 6A5        FIM R4 "60"; FIM R6 "10"; JMS TRANSLOOP;
0603 1707 06 6AB        FIM R4 "70"; FIM R6 "20"; JMS TRANSLOOP;
0604 1713 06 6B1        FIM R0 "10"; FIM R2 "80"; JMS SBRT;
0605 1719 06 6B7        FIM R0 "20"; FIM R2 "90"; JMS SBRT;
0606 1725 06 6BD        JMS CORDICAT;
0607 1727 06 6BF        FIM R2 "20"; FIM R6 "CC"; FIM R8 "00";
0608 1733 06 6C5        FIM R4 "10"; JMS DVRT;
0609 1737 06 6C9        FIM R0 "B0"; FIM R2 "00"; JMS ADRT;
0610 1743 06 6CF
0611 1743 06 6CF        REM
0612 1743 06 6CF        CONVERT CPA TIME TO PROPER HRS AND MINS AND STORE OUT;
0613 1743 06 6CF        FIM R4 "B0"; FIM R6 "C0"; JMS TRANSLOOP;
0614 1749 06 6D5        FIM R4 "10"; FIM R0 DAY; JMS FERT;
```

```
0615 1755 06 6DB         FIM R0 "00"; FIM R2 "10"; JMS SBRT;
0616 1761 06 6E1         RD0; RAR; JCN NZCY GOODTIME;
0617 1765 06 6E5         FIM R4 "00"; FIM R6 "B0"; JMS TRANSLOOP;
0618 1771 06 6EB         GOODTIME: FIM RA "B9"; FIM RC "EC";
0619 1775 06 6EF         LDM 14; XCH RF;
0620 1777 06 6F1         HROUT: SRC RA; RDM; XCH RE; LDM C; WRM;
0621 1782 06 6F6         LD RE; SRC RC; WRM; INC RB; INC RD; ISZ RF HROUT;
0622 1789 06 6FD         FIM RC "00"; JMS ZERORAM;
0623 1793 07 701         LDM 6; FIM RC "0A"; SRC RC; WRM;
0624 1798 07 706         FIM R2 "00"; FIM R4 "BC"; FIM R6 "10"; JMS MLRT;
0625 1806 07 70E         FIM RA "18"; FIM RC "EA"; LDM 13; XCH RF;
0626 1812 07 714         JMS TRANSPART;
0627 1814 07 716
0628 1814 07 716         REM COMPUTE TARGET COURSE AND SPEED;
0629 1814 07 716         FINDCANDS: FIM RA "E0"; SRC RA; RD3;
0630 1818 07 71A         JCN ZAC NOBACKROTATE2;
0631 1820 07 71C         FIM R4 "00"; FIM R0 MINPI2; JMS FERT;
0632 1826 07 722         FIM R0 "A0"; FIM R2 "0C"; JMS ADRT;
0633 1832 07 728         SRC R0;RD0;JCN ZCY NOBACKROTATE2;
0634 1836 07 72C         FIM R4 "00";FIM R0 TWOPI;JMS FERT;
0635 1842 07 732         FIM R0 "A0";FIM R2 "00";JMS ADRT;
0636 1848 07 738         NOBACKROTATE2: SRC RA; LDM 0; WR3;
0637 1851 07 73B         FIM RC "00"; FIM RA "F6";
0638 1855 07 73F         SRC RA; RD0; WRM; INC RB;
0639 1859 07 743         SRC RA; RD1; WRM; INC RB;
0640 1863 07 747         SRC RA; RD2; WRM; INC RB;
0641 1867 07 74B         SRC RA; RD3; WRM; FIM RA "F6";
0642 1872 07 750         JMS LOADBRG;
0643 1874 07 752         JMS CORDIC;
0644 1876 07 754         FIM RA "E0"; FIM RC "00"; JMS ZERORAM;
0645 1882 07 75A         FIM RC "0C";SRC RC;LDM 2;WRM;
0646 1887 07 75F         FIM RC "40"; JMS ZERORAM;
0647 1891 07 763         LDM 8; XCH RD;
0648 1893 07 765         SRC RA; RD0; SRC RC; WRM; INC RD;
0649 1898 07 76A         SRC RA; RD1; SRC RC; WRM; INC RD;
0650 1903 07 76F         SRC RA; RD2; SRC RC; WRM;
0651 1907 07 773         FIM R2 "40";FIM R4 "0C";FIM R6 "30";JMS MLRT;
0652 1915 07 77B         FIM R2 "30"; FIM R4 "10"; FIM R6 "60"; JMS MLRT;
0653 1923 07 783         FIM R2 "30"; FIM R4 "20"; FIM R6 "70"; JMS MLRT;
0654 1931 07 78B         FIM R4 "A0"; FIM R6 "CC"; JMS TRANSLOOP;
0655 1937 07 791         JMS CORDIC;
0656 1939 07 793         FIM R2 "C0"; FIM R4 "10"; FIM R6 "80"; JMS MLRT;
0657 1947 07 79B         FIM R2 "C0"; FIM R4 "20"; FIM R6 "90"; JMS MLRT;
0658 1955 07 7A3         FIM R4 "80"; FIM R6 "10"; JMS TRANSLOOP;
0659 1961 07 7A9         FIM R4 "90"; FIM R6 "20"; JMS TRANSLOOP;
0660 1967 07 7AF         FIM R0 "10"; FIM R2 "60"; JMS ADRT;
0661 1973 07 7B5         FIM R0 "20"; FIM R2 "70"; JMS ADRT;
0662 1979 07 7BB         JMS CORDICAT;
```

```
0663 1981 07 7BD
0664 1981 07 7BD          REM OUTPUT TARGET COURSE AND SPEED;
0665 1981 07 7BD          FIM RC "F0"; LDM 1; XCH RB; LDM 15; JMS DCRAM;
0666 1988 07 7C4          FIM RA "08"; FIM RC "F6"; LDM 12; XCH RF;
0667 1994 07 7CA          JMS TRANSPART;
0668 1996 07 7CC          FIM RC "10"; JMS ZERORAM;
0669 2000 07 7D0          LDM 5; FIM RC "15"; SRC RC; WRM;
0670 2005 07 7D5          FIM R2 "20"; FIM R4 "1C"; FIM R6 "30"; JMS MLRT;
0671 2013 07 7DD          FIM RA "38"; FIM RC "FC"; LDM 13; XCH RF;
0672 2019 07 7E3          JMS TRANSPART;
0673 2021 07 7E5          FIM RC "F2"; LDM 14; XCH RB; JMS FBACK;
0674 2027 07 7EB          ATLASTOHATLAST: JUN CPARETURN;
0675 2029 07 7ED          REM END MANEUVERING BOARD ALGORITHM;
0676 2029 07 7ED
0677 2029 07 7ED         .REM ROTATES POSITIONS PLUS 90 DEGREES;
0678 2029 07 7ED          REVOLVE:
0679 2029 07 7ED          FIM R4 "10"; FIM R6 "3C"; JMS TRANSLOOP;
0680 2035 07 7F3          FIM R4 "20"; FIM R6 "10"; JMS TRANSLOOP;
0681 2041 07 7F9          FIM R0 "10"; JMS CPLRT;
0682 2045 07 7FD          FIM R4 "30"; FIM R6 "2C"; JMS TRANSLOOP;
0683 2051 08 803          FIM R0 "70"; JMS CPLRT;
0684 2055 08 807          FIM R4 "70"; FIM R6 "6C"; JMS TRANSLOOP;
0685 2061 08 80D          FIM R4 "80"; FIM R6 "3C"; JMS TRANSLOOP;
0686 2067 08 813          FIM R4 "90"; FIM R6 "80"; JMS TRANSLOOP;
0687 2073 08 819          FIM R0 "8C"; JMS CPLRT;
0688 2077 08 81D          FIM R4 "30"; FIM R6 "90"; JMS TRANSLOOP;
0689 2083 08 823          LDM "F"; FIM R0 "E0"; SRC R0; WR3;
0690 2088 08 828          BBL 0;
0691 2089 08 829
0692 2089 08 829          REM ZEROES A RAM REGISTER AND ITS SIGN;
0693 2089 08 829          ZERORAM: CLB;
0694 2090 08 82A          ZRAM: SRC RC; WRM; ISZ RD ZRAM; WR0; BBL 0;
0695 2096 08 830
0696 2096 08 830          REM TRANSFERS BEARING FROM STORAGE TO
0697 2096 08 830              WORKING REGISTER;
0698 2096 08 830          LOADBRG:
0699 2096 08 830          JMS ZERORAM; LDM 8; XCH RD;
0700 2100 08 834          LDM 12; XCH RF; JMS TRANSPART; BBL 0;
0701 2105 08 839
0702 2105 08 839          REM TRANSFERS RANGE FROM STORAGE TO
0703 2105 08 839            WORKING REGISTER;
0704 2105 08 839          LOADRNG:
0705 2105 08 839          JMS ZERORAM; LDM 9; XCH RD;
0706 2109 08 83D          LDM 10; XCH RF; JMS TRANSPART; BBL 0;
0707 2114 08 842
0708 2114 08 842          REM TRANSFERS PART OF ONE REGISTER TO ANOTHER;
0709 2114 08 842          TRANSPART: SRC RA; RDM; SRC RC; WRM;
0710 2118 08 846          INC RB; INC RD; ISZ RF TRANSPART; BBL 0;
```

```
0711 2123 08 84B
0712 2123 08 84B            REM TRANSFERS MINUTES FROM STORAGE TO
0713 2123 08 84B                WORKING REGISTER;
0714 2123 08 84B            LOADTIMELO:
0715 2123 08 84B            JMS ZERORAM; LDM 8; XCH RC; LDM 13; XCH RF;
0716 2129 08 851            JMS TRANSPART; BBL 0;
0717 2132 08 854
0718 2132 08 854            REM TRANSFERS HOURS FROM STORAGE TO
0719 2132 08 854                WORKING REGISTER;
0720 2132 08 854            LOADTIMEHI:
0721 2132 08 854            LDM 9; XCH RC; LDM 14; XCH RF;
0722 2136 08 858            JMS TRANSPART; BBL 0;
0723 2139 08 85B
0724 2139 08 85B            REM LOADS PART OF REGISTER WITH CHARACTER
0725 2139 08 85B                IN ACCUMULATOR;
0726 2139 08 85B            DORAM:
0727 2139 08 85B            SRC RC; WRM; INC RD; ISZ RB DORAM; BBL 0;
0728 2145 08 861
0729 2145 08 861            REM CHANGES LEADING ZEROES TO BLANKS ON DISPLAY;
0730 2145 08 861            FBACK:
0731 2145 08 861            SRC RC; RDM; JCN NZAC NONZERO;
0732 2149 08 865            LDM "F"; WRM; LD RD; DAC; XCH RD; ISZ RB FBACK;
0733 2156 08 86C            NONZERO: BBL 0;
0734 2157 08 86D
0735 2157 08 86D            REM SETS UP SUBLOOP COUNT 7,9,9,10 FOR
0736 2157 08 86D                CORDIC ROUTINES AND FETCHES ARCTANGENT
0737 2157 08 86D                CONSTANTS FROM ROM;
0738 2157 08 86D            SETUP:
0739 2157 08 86D            LD RC;  RAR; JCN NZCY SETUP1;
0740 2161 08 871                    RAR; JCN NZCY SETUP2;
0741 2164 08 874                    RAR; JCN NZCY SETUP3;
0742 2167 08 877                    RAR; JCN NZCY SETUP4;
0743 2170 08 87A            FALLTHRU5: FIM R4 "50"; FIM R0 A5; JMS FERT;
0744 2176 08 880            LDM 6; XCH RE; BBL 12;
0745 2179 08 883            SETUP1: LDM 2; XCH RC;
0746 2181 08 885            FIM R4 "50"; FIM R0 A1; JMS FERT;
0747 2187 08 88B            LDM 15; XCH RE; BBL 0;
0748 2190 08 88E            SETUP2: LDM 4; XCH RC;
0749 2192 08 890            FIM R4 "50"; FIM R0 A2; JMS FERT;
0750 2198 08 896            LDM 9; XCH RE; BBL 15;
0751 2201 08 899            SETUP3: LDM 8; XCH RC;
0752 2203 08 89B            FIM R4 "50"; FIM R0 A3; JMS FERT;
0753 2209 08 8A1            LDM 7; XCH RE; BBL 14;
0754 2212 08 8A4            SETUP4: LDM 0; XCH RC;
0755 2214 08 8A6            FIM R4 "50"; FIM R0 A4; JMS FERT;
0756 2220 08 8AC            LDM 7; XCH RE; BBL 13;
0757 2223 08 8AF
0758 2223 08 8AF            SYN A=2,AP=3,CNT=3,R=4,RP=5,C=6,CP=7,G=8,GP=9,
```

```
2223 08 8AF     N=10,NP=11,K=12,KP=13,SHIFTA=14,SHIFTC=15;
2223 08 8AF
2223 08 8AF     REM DIVIDE ROUTINE, SETS UP TO USE DECDIV;
2223 08 8AF     DVRT:
2223 08 8AF     SRC A; RD0; RAR; JCN ZCY DV4;
2228 08 8B4     LD A; XCH R0; LDM 0; XCH R1; JMS CPLRT; STC;
2235 08 8BB     LDM 1; WR1;
2237 08 8BD     DV4: RAL; XCH RF; SRC C; RD0; RAR; JCN ZCY DV6;
2244 08 8C4     LD C; XCH R0; LDM 0; XCH R1; JMS CPLRT; STC;
2251 08 8CB     LDM 1; WR1;
2253 08 8CD     DV6: RAL; CLC; ADD RF; SRC Q; WR0;
2258 08 8D2     JMS DECDIV;
2260 08 8D4     CHCKPT:
2260 08 8D4     SRC A; RD1; JCN ZAC DV1;
2264 08 8D8     LD A; XCH R0; LDM 0; WR1; XCH R1; JMS CPLRT;
2271 08 8DF     DV1: SRC C; RD1; JCN ZAC DV2;
2275 08 8E3     LD C; XCH R0; LDM 0; WR1; XCH R1; JMS CPLRT;
2282 08 8EA     DV2: SRC Q; RD0; RAR; JCN ZCY ATLASTDV;
2287 08 8EF     CLC; RAL; WR0; LD Q; XCH R0; LDM 0; XCH R1;
2294 08 8F6     JMS CPLRT; ATLASTDV: BBL 0;
2297 08 8F9
2297 08 8F9     SYN A,AP,CNT,R,RP,C,CP,Q,QP,N,NP,K,KP,SHIFTA,SHIFTC;
2297 08 8F9
2297 08 8F9     BEGIN 2304;
2304 09 900
2304 09 900     REM CORDIC ROTATION MODE FOR SIN
2304 09 900          AND COS FUNCTIONS;
2304 09 900     CORDIC:
2304 09 900     FIM R4 "00"; FIM R6 "3C"; JMS TRANSLOOP;
2310 09 906     FIM R4 "40"; FIM R0 MINPI; JMS FERT;
2316 09 90C     FIM R0 "30"; FIM R2 "4C"; JMS ADRT;
2322 09 912     REM PLACE INPUT ANGLE IN FIRST OR FOURTH
2322 09 912          QUADRANTS;
2322 09 912     RD0; RAR; JCN NZCY SKIP1;
2326 09 916     FIM R4 "30"; FIM R0 MIN3PI2; JMS FERT;
2332 09 91C     FIM R0 "00"; FIM R2 "3C"; JMS ADRT;
2338 09 922     FIM R4 "20"; FIM R0 MINVK; JMS FERT;
2344 09 928     JUN JOIN1;
2346 09 92A     SKIP1: FIM R4 "30"; FIM R0 MINPI2; JMS FERT;
2352 09 930     FIM R0 "00"; FIM R2 "3C"; JMS ADRT;
2358 09 936     FIM R4 "20"; FIM R0 INVK; JMS FERT;
2364 09 93C     REM TEST INPUT ANGLE FOR DIRECTION  OF ROTATION
2364 09 93C          AND ROTATE PLUS OR MINUS 45 DEGREES;
2364 09 93C     JOIN1: FIM R4 "30"; FIM R0 A1; JMS FERT;
2370 09 942     FIM R0 "00"; SRC R0; RD0; RAR; JCN ZCY SKIP2;
2377 09 949     FIM R0 "00"; FIM R2 "3C"; JMS ADRT;
2383 09 94F     FIM R4 "20"; FIM R6 "1C"; JMS TRANSLOOP;
2389 09 955     JUN JOIN2;
```

```
0807 2391 09 957          SKIP2: FIM R0 "00"; FIM R2 "30"; JMS SBRT;
0808 2397 09 95D          FIM R4 "20"; FIM R6 "10"; JMS TRANSLOOP;
0809 2403 09 963          FIM R0 "10"; JMS CPLRT;
0810 2407 09 967          REM DO REST OF ROTATES;
0811 2407 09 967          JOIN2: LDM 2; XCH RC; LDM 12; XCH RA;
0812 2411 09 968          MLOOP: JMS SETUP; XCH RB;
0813 2414 09 96E          LOOP: FIM R0 "00"; SRC R0; RD0; RAR; JCN ZCY SKIPX;
0814 2421 09 975          LDM "F"; XCH RD; JUN JOINX;
0815 2425 09 979          SKIPX: LDM 0; XCH RD;
0816 2427 09 97B          JOINX: JMS ROTATE;
0817 2429 09 97D          ISZ RB LOOP;
0818 2431 09 97F          ISZ RA MLOOP;
0819 2433 09 981          REM END CORDIC ROTATION MODE,
0820 2433 09 981                RAM 10 HAS COS,RAM 20 HAS SIN;
0821 2433 09 981          ATLAST: BBL 0;
0822 2434 09 982
0823 2434 09 982
0824 2434 09 982          REM COMPLEMENTS REGISTER;
0825 2434 09 982          CPLRT:
0826 2434 09 982          CLC; COMPL: SRC R0; LDM 6; ADM; CMA; WRM;
0827 2440 09 988          ISZ R1 COMPL;
0828 2442 09 98A          STC; TENS: LDM 0; SRC R0; ADM; DAA; WRM; INC R1;
0829 2449 09 991          JCN NZCY TENS;
0830 2451 09 993          SRC R0; RD0; RAR; CMC; RAL; WR0; BBL 0;
0831 2458 09 99A
0832 2458 09 99A
0833 2458 09 99A          REM ADDS TWO REGISTERS INTO FIRST;
0834 2458 09 99A          ADRT: CLC;
0835 2459 09 99B          ADDR: SRC R0; RDM; SRC R2; ADM; DAA; SRC R0; WRM;
0836 2466 09 9A2          INC R3;
0837 2467 09 9A3          ISZ R1 ADDR;
0838 2469 09 9A5          RD0; XCH RF; SRC R2; RD0; ADD RF; SRC R0; WR0; BBL 0;
0839 2477 09 9AD
0840 2477 09 9AD
0841 2477 09 9AD          REM SUBTRACTS TWO REGISTERS, ANSWER IN FIRST;
0842 2477 09 9AD          SBRT:
0843 2477 09 9AD          LDM 10; XCH RF; CLB;
0844 2480 09 9B0          SUBT: SRC R0; RDM; SRC R2; SBM; JCN NZCY OK;
0845 2486 09 9B6          ADD RF; CLC;
0846 2488 09 9B8          OK: CMC; SRC R0; WRM; INC R3; ISZ R1 SUBT;
0847 2494 09 9BE          RD0; XCH RF; SRC R2; RD0; ADD RF; SRC R0; WR0; BBL 0;
0848 2502 09 9C6
0849 2502 09 9C6
0850 2502 09 9C6          REM TRANSFERS CONTENTS OF ONE REGISTER TO ANOTHER;
0851 2502 09 9C6          TRANSLOOP:
0852 2502 09 9C6          SRC R4; RDM; SRC R6; WRM; INC R5; ISZ R7 TRANSLOOP;
0853 2509 09 9CD          SRC R4; RD0; SRC R6; WR0; BBL 0;
0854 2514 09 9D2
```

```
0855 2514 09 9D2
0856 2514 09 9D2        BEGIN 2560;
0857 2560 10 A00
0858 2560 10 A00        REM CORDIC VECTORING MODE FOR ARCTAN AND
0859 2560 10 A00            MODULUS FUNCTIONS;
0860 2560 10 A00        CORDICAT:
0861 2560 10 A00        REM ZERO RAM FOR Z IN "00", X IN "10", Y IN "20";
0862 2560 10 A00        SYN A=2,AP=3,B=4,BP=5,C=6,CP=7,CEASE=14;
0863 2560 10 A00        FIM R0 "00"; CLB;
0864 2563 10 A03        ZEROLOOPAT: SRC R0; WRM; ISZ R1 ZEROLOOPAT; WR0;
0865 2568 10 A08        FIM R4 "20"; FIM R6 "30"; JMS TRANSLOOP;
0866 2574 10 A0E        SRC R6; RD0; RAR; JCN ZCY SWITCH1;
0867 2579 10 A13        FIM R0 "30"; JMS CPLRT;
0868 2583 10 A17        SWITCH1: FIM R4 "10"; FIM R6 "40"; JMS TRANSLOOP;
0869 2589 10 A1D        SRC R6; RD0; RAR; JCN ZCY SWITCH2;
0870 2594 10 A22        FIM R0 "40"; JMS CPLRT;
0871 2598 10 A26        REM TEST Y/X TO MINIMIZE TRUNCATION ERROR;
0872 2598 10 A26        SWITCH2: FIM R0 "30"; FIM R2 "40"; JMS SBRT;
0873 2604 10 A2C        SRC R0; RD0; RAR; JCN NZCY SWITCH3;
0874 2609 10 A31        FIM R4 "00"; FIM R0 MINPI2; JMS FERT;
0875 2615 10 A37        FIM R4 "10"; FIM R6 "30"; JMS TRANSLOOP;
0876 2621 10 A3D        FIM R0 "20"; JMS CPLRT;
0877 2625 10 A41        FIM R4 "20"; FIM R6 "10"; JMS TRANSLOOP;
0878 2631 10 A47        FIM R4 "30"; FIM R6 "20"; JMS TRANSLOOP;
0879 2637 10 A4D        REM ENSURE X POSITIVE;
0880 2637 10 A4D        SWITCH3: FIM R0 "10"; SRC R0; RD0; RAR;
0881 2642 10 A52        JCN ZCY SWITCH4;
0882 2644 10 A54        JMS CPLRT; FIM R0 "20"; JMS CPLRT;
0883 2650 10 A5A        FIM R4 "30"; FIM R0 MINPI; JMS FERT;
0884 2656 10 A60        FIM R0 "00"; FIM R2 "30"; JMS SBRT;
0885 2662 10 A66        REM TEST Y AND DRIVE TO 0,
0886 2662 10 A66            THEN DO REST OF ROTATES;
0887 2662 10 A66        SWITCH4: LDM 1; XCH RC; LDM 11; XCH RA;
0888 2666 10 A6A        MLOOPAT: JMS SETUP; XCH RB;
0889 2669 10 A6D        LOOPAT: FIM R0 "20"; SRC R0; RD0; RAR;
0890 2674 10 A72        JCN NZCY SKIPXAT;
0891 2676 10 A74        LDM "F"; XCH RD; JUN JCINXAT;
0892 2680 10 A78        SKIPXAT: LDM 0; XCH RD;
0893 2682 10 A7A        JCINXAT: JMS ROTATE;
0894 2684 10 A7C        ISZ RB LOOPAT;
0895 2686 10 A7E        ISZ RA MLOOPAT;
0896 2688 10 A80        REM X=X*1/K;
0897 2688 10 A80        CHECK:
0898 2688 10 A80        FIM R4 "30"; FIM R0 INVK; JMS FERT;
0899 2694 10 A86        FIM B "30"; FIM A "10"; FIM C "20"; JMS MLRT;
0900 2702 10 A8E        FIM R0 "00"; SRC R0; RD0; RAR; JCN ZCY ATLASTAT;
0901 2709 10 A95        FIM R4 "30"; FIM R0 TWOPI; JMS FERT;
0902 2715 10 A9B        FIM R0 "00"; FIM R2 "30"; JMS ADRT;
```

```
2721 10 AA1          REM END CORDIC VECTORING MODE,
2721 10 AA1              RAM 00 HAS ARCTAN,RAM 10 HAS MODULUS;
2721 10 AA1          ATLASTAT: BBL 0;
2722 10 AA2
2722 10 AA2          REM ROTATION ROUTINE FOR CORDIC ALGORITHMS;
2722 10 AA2          ROTATE:
2722 10 AA2          FIM R0 "00"; FIM R2 "50";
2726 10 AA6          LD RD; JCN ZAC SZ; JMS ADRT; JUN JZ;
2733 10 AAD          SZ: JMS SBRT; JZ:
2735 10 AAF          FIM R4 "10"; FIM R6 "30"; JMS TRANSLOOP;
2741 10 AB5          FIM R4 "20"; FIM R6 "40"; JMS TRANSLOOP;
2747 10 ABB          FIM R2 "40"; LDM 15; XCH R3; SRC R2; RDM; XCH R9;
2754 10 AC2          INC R3; LD RB; JCN ZAC NOSHIFTX; JMS SHIFT;
2760 10 AC8          NOSHIFTX: FIM R0 "10";
2762 10 ACA          LD RD; JCN ZAC SX; JMS ADRT; JUN JX;
2769 10 AD1          SX: JMS SBRT; JX:
2771 10 AD3          FIM R2 "30"; LDM 15; XCH R3; SRC R2; RDM; XCH R9;
2778 10 ADA          INC R3; LD RB; JCN ZAC NOSHIFTY; JMS SHIFT;
2784 10 AE0          NOSHIFTY: FIM R0 "20";
2786 10 AE2          LD RD; JCN ZAC SY; JMS SBRT; BBL "B";
2792 10 AE8          SY: JMS ADRT; BBL "A";
2795 10 AEB
2795 10 AEB          REM SHIFT FUNCTION FOR CORDIC ALGORITHMS;
2795 10 AEB          SHIFT:
2795 10 AEB          LD RB; XCH R9;
2797 10 AED          MINORLOOP: SRC R2; WRM; INC R3; ISZ RB MINORLOOP;
2802 10 AF2          LD R9; XCH RB; BBL 0;
2805 10 AF5
2805 10 AF5
2805 10 AF5          BEGIN 2816;
2816 11 B00
2816 11 B00          REM FETCHES CONSTANT FROM ROM;
2816 11 B00          FERT: LDM 8; XCH RF;
2818 11 B02          FETCHLOOP: FIN R2; XCH R2; SRC R4; WRM; INC R5;
2823 11 B07          XCH R3; SRC R4; WRM; INC R5; ISZ R1 NOINC;
2829 11 B0D          INC R0; NOINC: ISZ RF FETCHLOOP;
2832 11 B10          FIN R2; XCH R2; SRC R4; WR0; BBL 0;
2837 11 B15
2837 11 B15          REM CONSTANTS OF 360,-180,-270,-90 DEGREES;
2837 11 B15
2837 11 B15          TWOPI:
2837 11 B15          CON "000000000063000000";
2846 11 B1E          MINPI:
2846 11 B1E          CON "000000000028999910";
2855 11 B27          MIN3PI2:
2855 11 B27          CON "000000000037999910";
2864 11 B30          MINPI2:
2864 11 B30          CON "000000000019999910";
```

```
2873 11 B39
2873 11 B39        REM CONSTANTS OF 1/K,-1/K;
2873 11 B39        INVK:
2873 11 B39        CON "615495286000000000";
2882 11 B42        MINVK:
2882 11 B42        CON "484504713999999910";
2891 11 B4B
2891 11 B4B        REM ARCTAN CONSTANTS TABLE;
2891 11 B4B        A1:
2891 11 B4B        CON "000000000540000000";
2900 11 B54        A2:
2900 11 B54        CON "731395017500000000";
2909 11 B5D        A3:
2909 11 B5D        CON "896839275000000000";
2918 11 B66        A4:
2918 11 B66        CON "067592750000000000";
2927 11 B6F        A5:
2927 11 B6F        CON "875927500000000000";
2936 11 B78
2936 11 B78        REM TIME CONVERSION CONSTANTS;
2936 11 B78        DAY:
2936 11 B78        CON "000000000420000000";
2945 11 B81        HRSPERMIN:
2945 11 B81        CON "766666610000000000";
2954 11 B8A
2954 11 B8A        REM MULTIPLY ROUTINE;
2954 11 B8A        MLRT:
2954 11 B8A        CLB; XCH CP; LDM 0; XCH CBASE; LDM 0;
2959 11 B8F        ZEROLOOPM: SRC C; WRM; ISZ CP ZEROLOOPM; WR0;
2964 11 B94        LDM 4; XCH BP; SRC A; RD0; RAR; JCN ZCY ML4;
2971 11 B9B        LD A; XCH R0; LDM 0; XCH R1; JMS CPLRT; STC;
2978 11 BA2        LDM "F";SRC A;WR3;
2981 11 BA5        ML4: RAL; XCH RF; SRC B; RD0; RAR; JCN ZCY ML6;
2988 11 BAC        LD B; XCH R0; LDM 0; XCH R1; JMS CPLRT; STC;
2995 11 BB3        LDM "F";SRC B;WR3;
2998 11 BB6        ML6: RAL; CLC; ADD RF; SRC C; WR0;
3003 11 BBB        ML1: SRC B; RDM; XCH RF;
3006 11 BBE        ML2: LD RF; JCN ZAC ML3;
3009 11 BC1        DAC; XCH RF; LDM 5; XCH AP;
3013 11 BC5        LD CBASE; XCH CP; JMS MACRT; JUN ML2;
3019 11 BCB        ML3: INC CBASE; ISZ BP ML1;
3022 11 BCE        SRC C; RD0; RAR; JCN ZCY ML5; LDM 0; WR0; XCH R1;
3030 11 BD6        LD C; XCH R0; JMS CPLRT;
3034 11 BDA        ML5: SRC A;RD3;JCN ZAC ML8;
3038 11 BDE        LD A;XCH R0;LDM 0;XCH R1;JMS CPLRT;
3044 11 BE4        LDM 0;SRC A;WR3;
3047 11 BE7        ML8: SRC B;RD3;JCN ZAC ML7;
3051 11 BEB        LD B;XCH R0;LDM 0;XCH R1; JMS CPLRT;
```

```
0999 3057 11  BF1          LDM 0;SRC B;WR3;
1000 3060 11  BF4          ML7:BBL 0;
1001 3061 11  BF5
1002 3061 11  BF5          SYN A,AP,B,BP,C,CP,CBASE;
1003 3061 11  BF5
1004 3061 11  BF5          BEGIN 3072;
1005 3072 12  C00
1006 3072 12  C00          SYN A=2,AP=3,CNT=3,R=4,RP=5,C=6,CP=7,
1007 3072 12  C00          Q=8,QP=9,
1008 3072 12  C00          N=10,NP=11,K=12,KP=13,SHIFTA=14, SHIFTC=15;
1009 3072 12  C00
1010 3072 12  C00          MADRT:CLC;
1011 3073 12  C01          STMAD:SRC R2;RDM;SRC R6;ADM;DAA;WRM;
1012 3079 12  C07          ISZ R3 SKIPM1;BBL 0;
1013 3082 12  C0A          SKIPM1: ISZ R7 STMAD; BBL 0;
1014 3085 12  C0D
1015 3085 12  C0D          REM DECIMAL DIVISION ROUTINE,
1016 3085 12  C0D              WRITTEN BY
1017 3085 12  C0D              G. A. KILDALL,
1018 3085 12  C0D              ASSISTANT PROFESSOR,
1019 3085 12  C0D              NAVAL POSTGRADUATE SCHOOL,
1020 3085 12  C0D              MONTEREY, CALIFORNIA;
1021 3085 12  C0D          DECDIV: LDM 9; SRC A; WR2; SRC C; WR2; SRC Q; WR2;
1022 3092 12  C14          CLB; ZEROR: SRC Q; WRM; SRC R; WRM; INC RP;
1023 3098 12  C1A          ISZ QP ZEROR;
1024 3100 12  C1C          CLB; XCH NP; LOCATEZERO: LD NP; CMA; XCH AP; SRC A;
1025 3106 12  C22          RDM;
1026 3107 12  C23          JCN NZAC FOUNDZERO; ISZ NP LOCATEZERO;
1027 3111 12  C27          JUN ENDDIVIDE;
1028 3113 12  C29          FOUNDZERO: LD NP; XCH RP; CLB; XCH AP;
1029 3117 12  C2D          COPYA: SRC A; RDM; SRC R; WRM; INC AP;
1030 3122 12  C32          ISZ RP COPYA;
1031 3124 12  C34          LD NP; XCH SHIFTA; SRC A; RD2; ADD NP; XCH NP; TCC;
1032 3131 12  C3B          XCH N;
1033 3132 12  C3C          CLB; XCH KP;
1034 3134 12  C3E          LOCATEZERO1: LD KP; CMA; XCH CP; SRC C; RDM;
1035 3139 12  C43          JCN NZAC FOUNDZERO1; ISZ KP LOCATEZERO1; BBL 1;
1036 3144 12  C48
1037 3144 12  C48          FOUNDZERO1: LD KP; XCH SHIFTC; RD2; ADD KP; XCH KP;
1038 3149 12  C4D          TCC; XCH K;
1039 3151 12  C4F          SRC Q; RD2; ADD KP; XCH KP; LDM 0; ADD K; XCH K;
1040 3158 12  C56          CLC; LD KP; SUB NP; XCH QP; CMC; LD K; SUB N;
1041 3165 12  C5D          JCN NZCY NOUNDERF; BBL 0;
1042 3168 12  C60          NOUNDERF: JCN NZAC DOVERFLOW;
1043 3170 12  C62
1044 3170 12  C62          LDM 15; XCH NP; LD C; XCH N;
1045 3174 12  C66          COPYC1: SRC C; RDM; SRC N; WRM;
1046 3178 12  C6A          LD CP; JCN ZAC PASTCOPY1; DAC; XCH CP;
```

```
1047 3183 12 C6F           LD NP; DAC; XCH NP; JUN COPYC1;
1048 3188 12 C74
1049 3188 12 C74           PASTCOPY1: LD NP; JCN ZAC DIV;
1050 3191 12 C77           DAC; XCH NP; SRC N; LDM C; WRM;
1051 3196 12 C7C           JUN PASTCOPY1;
1052 3198 12 C7E           DIV: LDM 10; XCH K;
1053 3200 12 C80           SUB0: CLB; XCH CNT;
1054 3202 12 C82           SUB1: CLB; XCH RP; LD NP; XCH CP; SRC R;
1055 3207 12 C87           SUB2: RDM; SRC C; SBM; JCN NZCY COMPLEMENT;
1056 3212 12 C8C           ADD K; CLC;
1057 3214 12 C8E           COMPLEMENT: CMC; SRC R; WRM; INC RP;
1058 3218 12 C92           SRC R; ISZ CP SUB2;
1059 3221 12 C95           LD RP; JCN ZAC CHECKCARRY;
1060 3224 12 C98           RDM; SUB CP; WRM; CMC;
1061 3228 12 C9C           CHECKCARRY: JCN NZCY CARRYOUT;
1062 3230 12 C9E           INC CNT; JUN SUB1;
1063 3233 12 CA1           CARRYOUT: LD NP; XCH CP; CLB; XCH RP;
1064 3237 12 CA5           ADD4: SRC C; RDM; SRC R; ADM; DAA; WRM; INC RP;
1065 3244 12 CAC           ISZ CP ADD4;
1066 3246 12 CAE           LD RP; JCN ZAC SKIPADD; TCC; SRC R; ADM; WRM;
1067 3253 12 CB5           SKIPADD: SRC Q; LD CNT; WRM; LD QP; JCN ZAC ENDDIVIDE;
1068 3259 12 CBB           DAC; XCH QP; ISZ NP SUB0;
1069 3263 12 CBF
1070 3263 12 CBF           ENDDIVIDE: CLB; XCH NP; LD SHIFTC; XCH CP;
1071 3267 12 CC3           COPYC2: SRC C; RDM; SRC N; WRM; INC NP;
1072 3272 12 CC8           ISZ CP COPYC2;
1073 3274 12 CCA           LD NP; JCN ZAC PASTFILL;
1074 3277 12 CCD           FILLZ: SRC N ; CLB; WRM; ISZ NP FILLZ;
1075 3282 12 CD2           PASTFILL: BBL 0;
1076 3283 12 CD3           DOVERFLOW: BBL 1;
1077 3284 12 CD4           SYN A,AP,CNT,R,RP,C,CP,Q,QP,N,NP,K,KP,SHIFTA,SHIFTC;
1078 3284 12 CD4           END
0   ERRORS.
```

LIST OF REFERENCES

1. Tinklepaugh, N., and Roth, A. "CPA Calculator." System description prepared at Naval Electronics Laboratory Center, San Diego, Ca., 1972. (Photocopied.)

2. Tinklepaugh, N., and Roth, A. "Closest Point of Approach Calculator." Demonstration materials prepared at Naval Electronics Laboratory Center, San Diego, Ca., 1972. (Mimeographed.)

3. Fritts, D. E. "Device Description, Scientific Read Only Memory (P/N 15171NA)." Specification prepared at North American Rockwell Microelectronics Company, Anaheim, Ca., October 25, 1971. (Photocopied.)

4. North American Rockwell Microelectronics Company. "Functional Description: PCPU Sample Calculator." Product bulletin, Anaheim, Ca., October, 1971. (Photocopied.)

5. ____________. Telephone interview with N. Tinklepaugh and A. Roth, Naval Electronics Laboratory Center, San Diego, Ca. April 5, 1973.

6. Dutton, Benjamin. _Navigation and Nautical Astronomy_. 9th ed. revised. Annapolis: United States Naval Institute, 1948.

7. Kildall, Gary A. "A Chebyshev Approximation Routine for the Intel MCS-4 Computer." Monterey, Ca.: Computer Science Group, United States Naval Postgraduate School, September, 1972. (Mimeographed.)

8. Volder, Jack E. "The CORDIC Trigonometric Computing Technique," _IRE Transactions on Electronic Computers_, Vol. EC-8, No. 3 (September, 1959), pp. 330-334.

9. Walther, J. S. "A Unified Algorithm for Elementary Functions," _Spring Joint Computer Conference_, Vol. 38 (May, 1971), pp. 379-385.

10. Schmid, Herman, and Bogacki, Anthony. "Use decimal CORDIC for generation of many transcendental functions," _EDN_, Vol. 18, No. 4 (February 20, 1973), pp. 64-73.

11. Walther, J. S. Letter to Dr. Richard Franke, United States Naval Postgraduate School, Monterey, Ca., from Hewlett-Packard Laboratories, Palo Alto, Ca., February 13, 1973.

12. Kildall, Gary A. "An MCS-4 Assembler and Interpreter for the IBM System/360." Monterey, Ca.: Computer Science Group, United States Naval Postgraduate School, July, 1972. (Mimeographed.)

13. Intel Corporation. _MCS-4 Micro Computer Set: Users Manual_. Rev. 3. Santa Clara, Ca.: Intel Corporation, July, 1972.

14. Brubaker, Ray H., Jr. "I/O Simulation using the MCS-4 Interpreter for the IBM System/360." Monterey, Ca.: Computer Science Group, United States Naval Postgraduate School, July, 1972. (Mimeographed.)

INITIAL DISTRIBUTION LIST

		No Copies
1.	Defense Documemtation Center Cameron Station Alexandria, Virginia 23314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chairman, Computer Science Group, Code 72 Naval Postgraduate School Monterey, California 93940	1
4.	Asst Professor G. A. Kildall, Code 53 Kd Naval Postgraduate School Monterey, California 93940	1
5.	Inst Ray H. Brubaker, Jr., Code 53 Bh Naval Postgraduate School Monterey, California 93940	1
6.	Asst Professor R. Franke, Code 53 Fe Naval Postgraduate School Monterey, California 93940	1
7.	Mr. Norm Tinklepaugh, Code 4300 Naval electronics laboratory Center San Diego, California 92152	1
8.	LCDR K. H. Kerns 128 Belle Drive Marina, California 93933	1
9.	LT R. S. Cooper 2200 South Ocean Boulevard Apartment 608 Delray Beach, Florida 33444	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A MICROCOMPUTER SOLUTION TO MANEUVERING BOARD PROBLEMS

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*
Master's Thesis; June 1973

5. AUTHOR(S) *(First name, middle initial, last name)*
Kenneth Harper Kerns
Roger Stuart Cooper

6. REPORT DATE June 1973	7a. TOTAL NO. OF PAGES 83	7b. NO. OF REFS 14
8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A special-purpose digital computer has been developed using the new MOS LSI microprocessor technology. The primary goal of this work was to solve a fairly complicated task using a minimal amount of random logic and limited development time. This computing system solves ships' maneuvering board problems including the determination of course, speed, and closest point of approach of other ships. Ten contacts can be tracked simultaneously. The triangulation involved in these calculations is performed using a decimal version of the CORDIC algorithm. A complete maneuvering situation can be computed in 4.5 seconds. Except for hardware required to drive the displays, all logic is contained in the software-encoded read-only-memory chips, which drive a single CPU chip. Approximately eight hundred man-hours were required to develop the programs and prototype the hardware. The system is modular and easy to maintain, low in bulk, power consumption, and cost.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6811

A-31408

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
CORDIC algorithm						
CPA (closest point of approach)						
LED (light-emitting diode) display						
maneuvering board						
MCS-4 Micro Computer Set						
microcomputer						
microprocessor						
PROM (programmable read-only-memory)						
RAM (random-access-memory)						
relative movement problem						
ROM (read-only-memory)						

DD FORM 1 NOV 65 1473 (BACK)
S/N 0101-807-6821

Thesis 145316
K3875 Kerns
c.1 A microcomputer solution to maneuvering board problems.